**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

**КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

# ВИПУСКНА РОБОТА

**на тему:**

## «Віртуальне середовище для вивчення методик побудови безпечних VPN-з'єднань»

**Завідувач**
**випускаючої кафедри** **Довбиш А.С.**

**Керівник роботи** **Лаврик Т.В.**

**Студента групи КБ – 71** **Басова М.В.**

**СУМИ 2021**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Затверджую_____
Зав. кафедрою Довбиш А.С.
"_____" _____2021 г.

# ЗАВДАННЯ

**до випускної роботи**

Студента четвертого курсу, групи КБ-71 спеціальності "Кібербезпека" денної форми навчання Басова Максима Вікторовича.

**Тема:** "**Віртуальне середовище для вивчення методик побудови безпечних VPN-з'єднань**"

Затверджена наказом по СумДУ
№_____ от _____2021 г.

**Зміст пояснювальної записки**: 1) аналітичний огляд принципів організації технології VPN; 2) опис основних протоколів для організації VPN та їх порівняльна характеристика; 3) вибір та опис основного програмного забезпечення для реалізації віртуального захищеного середовища; 4) розробка віртуального захищеного середовища та тестування коректності роботи мережі на всіх пристроях в системі.

Дата видачі завдання "_____"_____2021 р.
Керівник випускної роботи _____ Лаврик Т.В.
Завдання приняв до виконання _____ Басов М.В.

# РЕФЕРАТ

**Записка:** 49 стор., 41 рис., 2 табл., 1 додаток, 10 джерел.

**Об’єкт дослідження** — віртуально захищене середовище для передачі даних між приватними мережами через захищені канали трансляції інформації поверх публічних ліній передачі даних.

**Мета роботи** — розроблення віртуального захищеного середовища на основі протоколів, які задіяні в організації VPN мережі, налаштування системи клієнт-серверної архітектури та всіх допоміжних пристроїв, що задіяні в реалізації.

**Методи дослідження** — метод аналітичного огляду, метод порівняння та аналогій, метод моделювання.

**Результати** — розроблено віртуально захищене середовище із використанням операційної системи Linux, а саме дистрибутиву Centos 8 та утиліт, які були задіяні для реалізації OpenVPN сервера, клієнта та налаштування конфігураційних файлів. Це дало змогу об’єднати локальні мережі в єдину захищену мережу для безпечної передачі даних.

ВІРТУАЛЬНЕ СЕРЕДОВИЩЕ, VPN-МЕРЕЖА, ПУБЛІЧНА ТА ПРИВАТНА МЕРЕЖА, БРАНДМАУЕР, ЦЕНТР СЕРТИФІКАЦІЇ, LINUX, ШЛЮЗИ, VPN-ТУНЕЛЬ, КОНФІГУРАЦІЙНІ ФАЙЛИ, МЕРЕЖЕВІ НАЛАШТУВАННЯ.

# ЗМІСТ

# ВСТУП

Сьогодні в період діджиталізації одним із головних пунктів є надійний захист в інформаційному просторі інформаційного забезпечення та попередження спотворення, знищення, несанкціонованої модифікації, зловмисного отримання і використання інформації. У час розвитку новітніх технологій популярною тенденцією на більшості підприємств та великих компаній є перехід офісних працівників на віддалений режим роботи. Сьогодні корпоративні мережі є невід'ємною частиною сучасних компаній. За допомогою таких мереж можна безпечно передавати і отримувати інформацію. Даний підхід є дуже зручним, оскільки персонал не прив'язаний до конкретного місця, а може працювати з будь-якого місця планети. Однак канали передачі інформації та комп'ютери працівників стають більш вразливими для зловмисників, які хочуть заволодіти та скористатися інформацією та призвести до грошових втрат, заманювання клієнтів до своєї організації або взагалі до банкрутства. Для того, щоб забезпечити належний захист інформації каналів передачі інформації рекомендують використовувати так звані віртуальні приватні мережі (VPN) для захисту інтернет підключення до вашої корпоративної мережі, а також протидії кіберзагрозам.

Основними перевагами технології VPN є доволі дешеве впровадження, в подальшому основні затрати будуть зводитися до оплати послуг Інтернет провайдера, можливість підключення великої кількості абонентів, які знаходяться в різних кутках світу і звісно ж безпека передачі даних. Саме завдяки гнучкості та економічності, VPN активно витісняють LAN з ринку, оскільки до основного недоліку даної технології можна віднести високу вартість впровадження та неможливість підключення віддаленого персоналу. Затрати на використання та обслуговування VPN майже в три рази нижчі від логічних структур побудованих за технологією LAN.

# 1. ІНФОРМАЦІЙНИЙ ОГЛЯД

## 1.1 Основні принципи організації технології VPN

VPN (Virtual Private Network) — це віртуальна приватна мережа, яка є об’єднанням приватних систем та звичайних робочих комп’ютерів через публічне зовнішнє середовище трансляції інформації в єдину цілісну мережу, що забезпечує належний рівень захисту даних [1].

Створення віртуальної приватної мережі відбувається шляхом побудови віртуальних захищених каналів зв’язку, які створюються на базі відкритих каналів зв’язку загальнодоступної мережі. Ці віртуальні захищені канали зв’язку називаються тунелями VPN.

В основному технологія VPN використовується для об'єднання, наприклад, центрального офісу компанії з її філіалом, офісом бізнес-партнерів і віддалених користувачів для того, щоб безпечно обмінюватися інформацією через Інтернет. Також VPN використовується для забезпечення безпечного доступу до сервісів у внутрішній мережі. Це потрібно для того, щоб тільки користувачі внутрішньої мережі мали доступ до неї [2].

Основною метою технології тунелювання є інкапсуляція інформації, тобто упакування пакетів, що передаються разом зі службовими полями в новостворений пакет уже з іншими службовими полями. Слід відзначити, що сама технологія тунелювання не забезпечує надійних захист до закритої інформації, але саме завдяки тунелюванню з'являється можливість повного криптографічного захисту інкапсульованих пакетів. Для того, щоб забезпечити основні властивості інформаційної безпеки, користувач зашифоровує пакети, інкапсулює їх у зовнішній пакет уже з новим IP-заголовком і надсилає його через допоміжну мережу (рис. 1.1).

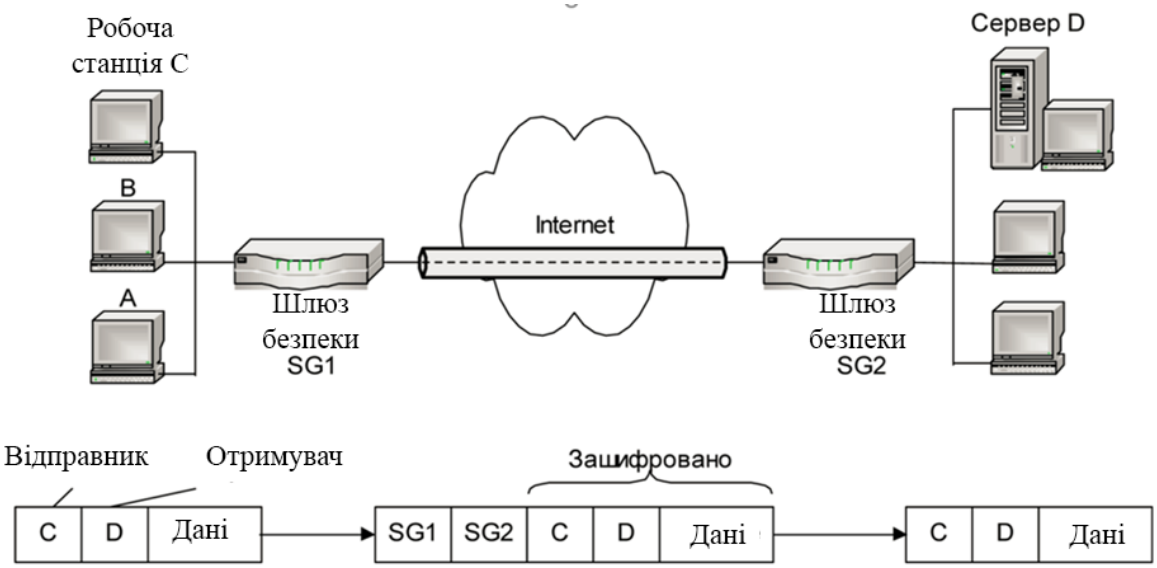


Рисунок 1.1 – Схема захищеного тунелю

Особливістю тунелювання є те, що дана технологія дозволяє зашифрувати інкапсульований пакет разом із усіма полями заголовка, а не тільки поле, в якому знаходяться дані. Шифрування є дуже важливим етапом, оскільки в заголовці пакета є поля, за допомогою яких зловмисники можуть отримати важливу інформацію. Дані з заголовку інкапсульованого пакета можуть містити інформацію про локальну структуру системи – дані про кількість підмереж і комп'ютерів та їх ір-адреси. Використовуючи такі дані зловмисник може організувати атаку на корпоративну мережу.

Пристрої VPN можуть відігравати у віртуальних приватних системах функцію VPN-клієнта, VPN-сервера або вузла безпеки VPN.

VPN-клієнт представляє собою програмний або програмно-апаратний комплекс, що реалізується в більшості випадків на базі користувацького комп'ютера. За допомогою спеціального програмного забезпечення виконується шифрування й аутентифікація трафіку в системі, це потрібно для того, щоб безпечно обмінюватися інформацією з усіма вузлами в приватній мережі.

Першочерговою метою VPN-серверу є забезпечення конфіденційності інформації компанії або звичайного користувача, а також побудові захищених з'єднань з персональними комп'ютерами і сегментами локальних систем.

Вузол безпеки VPN в свою чергу потрібен для того, щоб шифрувати та виконувати автентифікацію хостів. Розміщувати шлюз безпеки потрібно таким чином, щоб весь трафік який повинен надходити до корпоративної мережі проходив через нього[2].

## 1.2 Порівняльна характеристика протоколів для організації VPN

### 1.2.1. Протокол PPTP

PPTP (Point-to-Point Tunneling Protocol) – є одним із найдавніших VPN протоколів. PPTP – це тунельний протокол типу точка-точка, за рахунок якого можна реалізувати захищене з'єднання з сервером використовуючи тунелювання у звичайній локальній мережі.

Своєї популярності даний тунельний протокол набув за рахунок компанії Microsoft. Дана корпорація включила в усі версії Microsoft Windows починаючи з Windows 95 OSR2, включаючи до складу операційної системи PPTP-клієнт.

Для роботи PPTP протокол використовує два з'єднання, так само як протокол FTP, але має свої відмінності. Перше з'єднання працює за рахунок використання протоколу TCP на порту 1723, а друге з'єднання за рахунок протоколу GRE. Протокол GRE є розробкою компанії Cisco Systems, основним завданням якого є інкапсуляція пакетів мережевого рівня моделі OSI в IP-пакети. Для того, щоб користувачі, які знаходилися за технологією NAT могли з'єднуватися з сервером, протокол GRE був вдосконалений, а саме був доданий заголовок Call ID за допомогою якого маршрутизатори могли відслідковувати сеанс зв'язку між клієнтом локальної мережі до серверу який знаходився за технологією NAT і навпаки. Дана технологія використовується майже на всіх клієнтських пристроях.

З плином часу було виявлено, що протокол PPTP є не зовсім безпечним, оскільки після обриву сеансу зв’язку, для його відновлення потрібен не малий проміжок часу. Також в 2012 році була знайдена вразливість в даному протоколі, яка дозволяла розшифровувати дані за два дні, яку компанія Microsoft швидко виправила, але значною перевагою даного протоколу залишається швидкість передачі даних.

### 1.2.2. Протокол IPSec

IPSec або повна назва IP Security – це набір протоколів, основним завданням яких є шифрування, автентифікація та забезпечення цілісності даних при передачі IP-пакетів (рис. 1.2) [9].

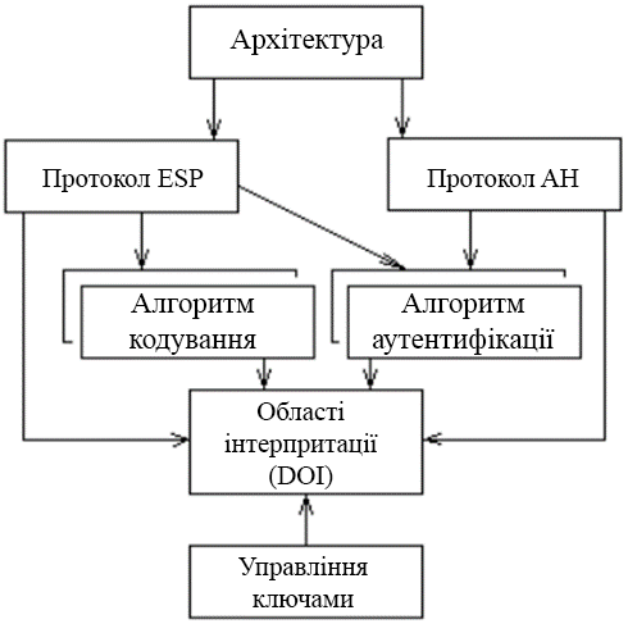


Рисунок 1.2 – Архітектура IPSec

IP Authentication Header (AH) забезпечує цілісність без встановлення з’єднання, автентифікацію та додаткову функцію захисту від повторів пакетів.

AH використовує хеш-алгоритм для обчислення значення хеша як для корисного навантаження, так і для заголовка пакета, забезпечуючи цілісність пакета. Однак це має негативний наслідок, оскільки AH не буде працювати через пристрої, в яких реалізована функція NAT, оскільки NAT змінить IP-заголовок, а сам хеш не зміниться. Пристрій, на який прийде даний пакет буде

вважати, що пакет був модифікований під час передачі, а отже його буде відхилено.

ESP (Encapsulating Security Payload) протокол також забезпечує такі три основні функції як конфіденційність, автентифікація і цілісність. Даний протокол є більш безпечним та ефективним, ніж AH оскільки хешування, яке використовується для цілісності даних, не включає в себе IP-заголовок, а отже ESP буде працювати з пристроями, які реалізують технологію NAT [3].

### 1.2.3. Протокол L2TP/IPSec

Layer 2 Tunneling Protocol або L2TP сьогодні є майже у всіх новостворених операційних системах і працює з усіма гаджетами, які здатні працювати з технологію VPN. L2TP – протокол тунелювання другого рівня. Даний протокол є розширенням протоколу PPTP та містить в собі найкращі функції, які були в протоколі PPTP від Microsoft і L2F від Cisco Systems, оскільки ці компанії займалися розробкою L2TP протоколу.

L2TP не здатний шифрувати трафік, який проходить через нього, тому в основному його використовують в парі з IPSec. Зв’язка L2TP/IPSec вважається безпечнішою, ніж PPTP, оскільки в L2TP/IPSec використовуються два види шифрування, а саме 3DES та AES, але в більшості випадків використовується AES.

Використання протоколів у парі має свої недоліки. Одним з цих недоліків є подвійне розміщення даних в пакетах, яке погано впливає на ефективність передачі даних. Також L2TP використовує 500 UDP-порт, який часто блокується брандмауером.

Основними компонентами L2TP є L2TP Access Concentrator, завданням якого є завершення сеансу, та мережевим сервером L2TP Network Server (LNS), який є завершальним пристроєм і відповідає за автентифікацію трафіку (рис. 1.3) [7].

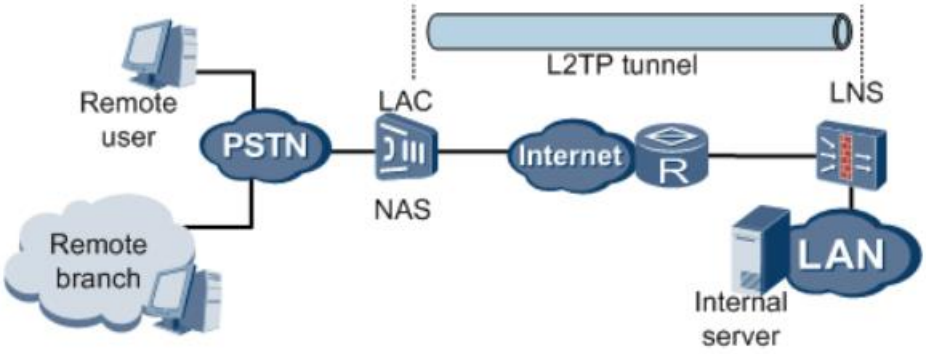


Рисунок 1.3 – Типова мережа на основі L2TP

### 1.2.4. Протокол OpenVPN

OpenVPN – протокол з відкритим вихідним кодом, який використовує бібліотеки Open SSL, TLS і ряд інших технологій. Сьогодні даний протокол є першочерговим для використання у корпоративній сфері VPN-сервісами і реалізується на всіх доступних платформах через допоміжне програмного забезпечення.

Перевагою протоколу OpenVPN є його різноманітність в налаштуванні. Його можна налаштувати на будь-якому порту, що дає змогу замаскувати трафік наприклад під HTTPS, якщо буде обраний 443 порт для налаштування, що тільки ускладнить процес для зловмисників.

Однак гнучкість даного протоколу призводить до таких складностей, як завантаження та налаштування конфігураційних файлів, але деякі провайдери можуть забезпечити уже готовими VPN-клієнтами, що значно полегшить процес використання VPN.

Також одним із переваг протоколу VPN можна вважати регулярні аудити безпеки. На сьогоднішній момент протокол OpenVPN вважається одним із самих надійних, який забезпечує швидку передачу даних.

Розглянемо відмінності проаналізованих протоколів, які задіяні у реалізації VPN технології (табл. 1).

Таблиця 1 – Порівняльна характеристика протоколів

| | PPTP | L2TP/IPSec | OpenVPN |
|---|---|---|---|
| Підтримка ОС | Windows, macOS, IOS, деякий час була підтримка GNU/Linux. | Windows, Mac OS X, Linux, iOS, Android. Більшість операційних систем не потребують додаткового ПО. | Solaris, OpenBSD, FreeBSD, NetBSD, GNU/Linux, Apple Mac OS X, QNX, Microsoft Windows, Android, iOS. |
| Шифрування | Microsoft Point-to-Point Encryption (MPPE), в якому реалізовано RSA RC4 з 128-бітними сеансовими ключами | 3DES або AES | За замовчування AES-256-CBC |
| Порти | TCP-порт 1723 | Для обміну ключами – UDP-порт 500, UDP-порт 1701 – для конфігурування, UDP-порт 5500 для обходу NAT | Будь-який доступний |
| Недоліки | RSA RS4 є вразливим до атаки Bit flipping | Шифр 3DES вразливий до атаки meet-in-the-middle | Прогалин в безпеці не було виявлено |

## 1.3 Постановка задачі

Аналіз існуючих протоколів, які використовуються для побудови безпечних VPN-з’єднань дозволяє дійти висновку, що ці протоколи відрізняються між собою, мають свою специфіку та застосовуються для вирішення різних задач.

На основі цього було вирішено розробити віртуальне захищене середовище, за допомогою якого будуть розглянуті окремі методики побудови VPN-з’єднань. Це дозволить досягнути вирішення задач інформаційної безпеки, а саме забезпечення доступності, цілісності, конфіденційності особистих та корпоративних даних, до яких має доступ лише обмежене коло людей.

Метою даної роботи є розробка віртуального середовища для вивчення методик побудови безпечних VPN-з’єднань. Це віртуальне середовище може бути використане для таких задач:

- оцінка основних протоколів для реалізації VPN-з’єднань;
- вибір операційної системи та програмних додатків для реалізації VPN-з’єднань та захищеного середовища;
- коректне налаштування основних компонентів мережі для реалізації захищеного середовища в корпоративній мережі;
- вивчення основних методів побудови захищених з'єднань у мережі.

# 2. ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ ТА ПІДГОТОВКА СИСТЕМИ ДО НАЛАШТУВАННЯ

## 2.1 Вибір основного програмного забезпечення для реалізації VPN

Для розробки віртуального середовища було обрану операційну систему Linux, оскільки більшість експертів вважають, що саме за допомогою неї можна забезпечити належний захист даних, файлів конфігурації та іншої важливої інформації, а також Linux найпопулярніша ОС на серверах. Належний захист ядра системи, усіх служб та програмних додатків є також одним з основних пунктів, які роблять дану систему безпечною з точки зору захисту даних[4].

Під час вибору дистрибутиву Linux, було обрано для реалізації Centos 8, тому, що за допомогою даної операційної системи можливо завантажити та налаштувати POPTOP або PPTPD сервер – це реалізація серверу на основі протоколу PPTP та OpenVPN серверу, вихідний код якого є загальнодоступним для майже всіх операційних систем. Також значною перевагою даного дистрибутиву є велика кількість репозиторіїв, які містять велику кількість безкоштовних програмних додатків та уже налаштованих файлів конфігурацій.

Для того, щоб завантажити та налаштувати сервер на основі протоколу PPTP необхідно додати додатковий репозиторій та завантажити відповідні пакети:

yum -y install epel-release;

yum -y install ppp pptpd net-tools iptables-services;

Основним файлом конфігурації даного серверу є файл, який знаходиться в /etc/pptpd.conf.

Щоб завантажити та почати налаштування OpenVPN серверу також необхідно додати додатковий репозиторій, аналогічний що і для серверу на основі протоколу PPTP та завантажити нижче зазначені пакети:

yum -y install epel-release;

yum -y install openvpn;

Також знадобиться утиліта Easy-RSA, для того щоб згенерувати сертифікати, які потім будуть застосовані в подальшому:

wget https://github.com/OpenVPN/easy-rsa/archive/master.zip

Основними для конфігурування є такі файли:

- /etc/openvpn/server.conf – конфігураційний файл сервера;
- /etc/openvpn/ccd/client – налаштування для кожного клієнта;
- /var/log/openvpn/openvpn.log та /var/log/openvpn/openvpn.log – файли в які будуть занотовуватися основні події, які виконуються на стороні сервера та клієнта.

## 2.2 Вибір програмного забезпечення для віртуалізації

Сьогодні існує безліч програмних додатків для віртуалізації, але деякі з них є комерційним продуктом. Одним з таких додатків є VMware Workstation вихідний код якого є закритим. Даний продукт більше підходить для розробників ПЗ або інженерів, які займаються тестуванням, оскільки вона має велику кількість дрібниць, що використовуються кожного дня і яких немає в інших додатках. У нашому випадку більше підходить продукт VirtualBox від компанії Oracle. Даний додаток є безкоштовним, вихідний код якого є відкритим. Перевагами даного продукту є:

- підтримка великої кількості операційних систем, а саме Windows, Linux, Mac OS X и Solaris;
- можливість обмеження споживання ресурсів CPU і введення-виведення;
- можливість регулювання відеопам’яті;
- зрозумілий графічний інтерфейс;
- зручний редактор мережевої взаємодії на хості та інше.

## 2.3 Встановлення та налаштування усіх сегментів мережі

Для розробки віртуального середовища було використано програму VirtualBox, оскільки дана утиліта є зручною у використанні та є безкоштовною. Віртуальне середовище складається із трьох сегментів мережі, двох локальних мереж, локальна мережа за серверною частиною та локальна мережа за клієнтською частиною, а також мережа між шлюзами. В таблиці представлено імена комп'ютерів та їхні IP-адреси (Таблиця 2).

Таблиця 2 – Таблиця налаштувань IP-адрес

| Комп'ютер | IP-адреса |
|---|---|
| Server | 192.168.1.12/24 |
| | 10.0.9.3/24 |
| Client | 192.168.1.14/24 |
| | 10.0.10.3/24 |
| PC1 | 10.0.9.4/24 |
| PC2 | 10.0.10.4/24 |

На серверах та на звичайних комп'ютерах встановлено операційну систему Centos 8. Мінімальні системні вимоги до цієї операційної системи невеликі, а саме 1 ГБ оперативної пам'яті, одноядерний процесор та 10 ГБ вільного місця на жорсткому диску. Для роботи було створено 4 віртуальних машини (рис. 2.1). Фізична топологія мережі має такий вигляд (рис. 2.2).

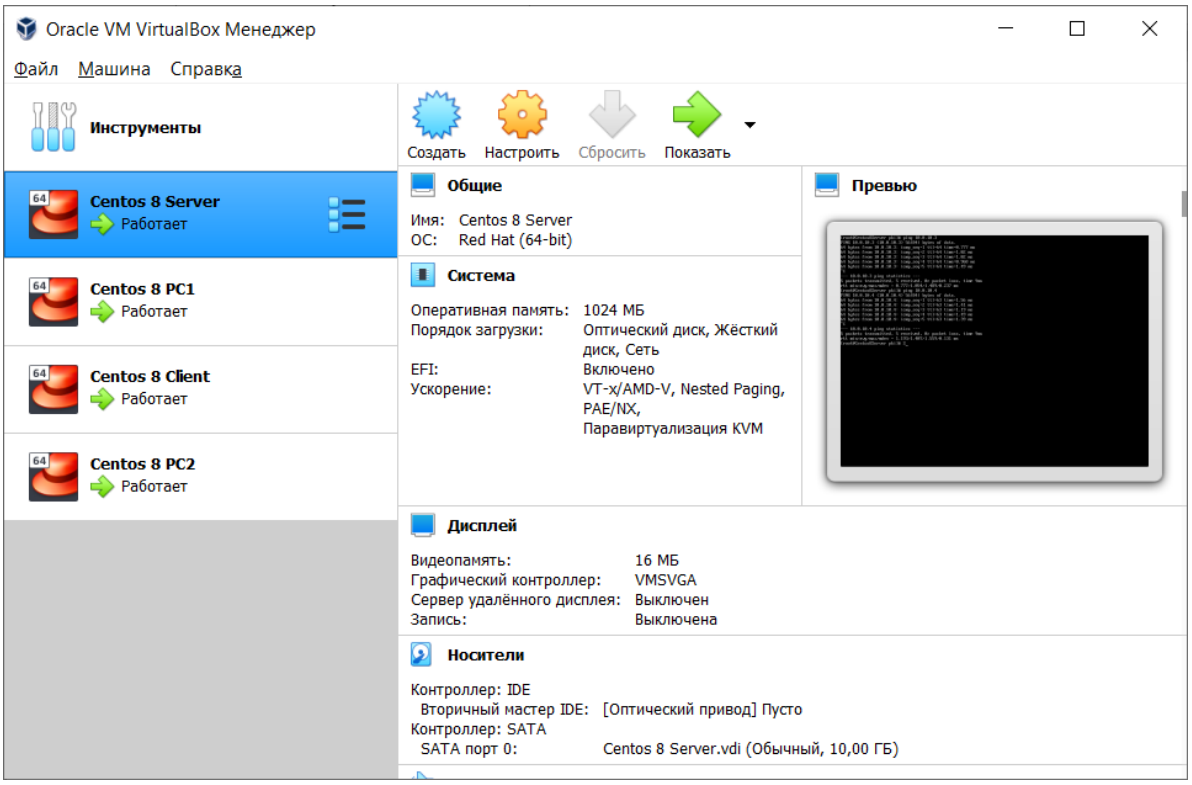


Рисунок 2.1 – Створені віртуальні машини

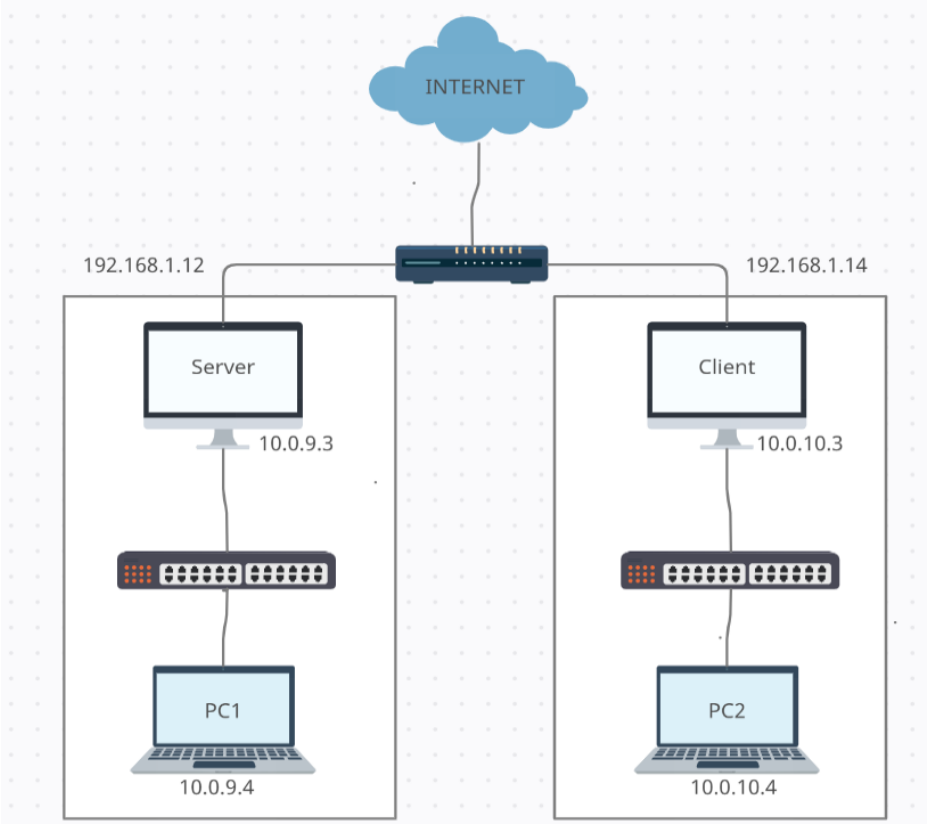


Рисунок 2.2 – Фізична топологія мережі

## 2.4 Оновлення елементів системи та завантаження допоміжних утиліт

Першим кроком після встановлення Centos 8 на наш ПК необхідно оновити систему за допомогою команди - dnf update. Це потрібно для того, щоб оновити системні файли, додатки, ядро системи та багато іншого важливого інструментарію(рис. 2.3, рис. 2.4) [7].

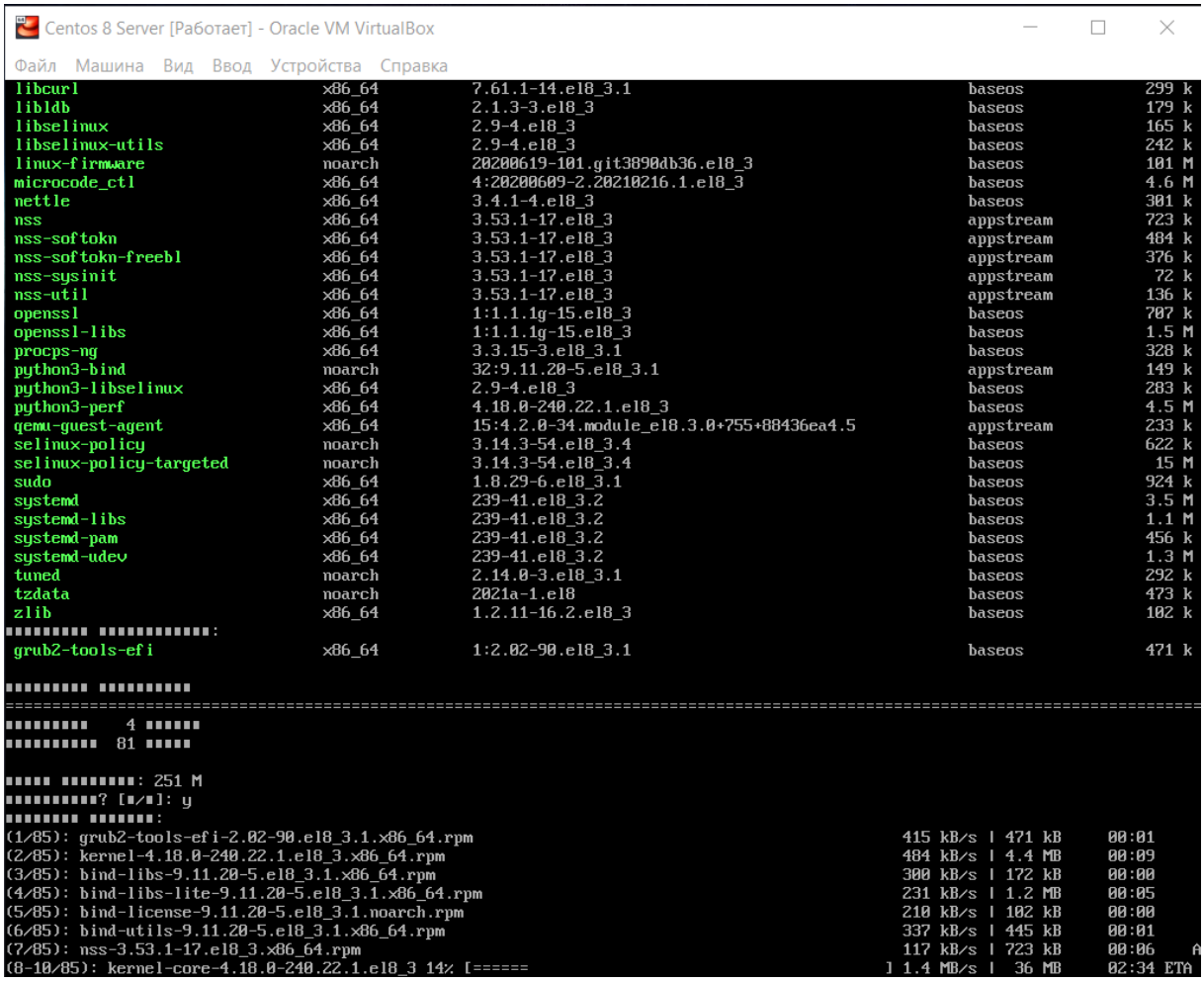


Рисунок 2.3 – Оновлення системи

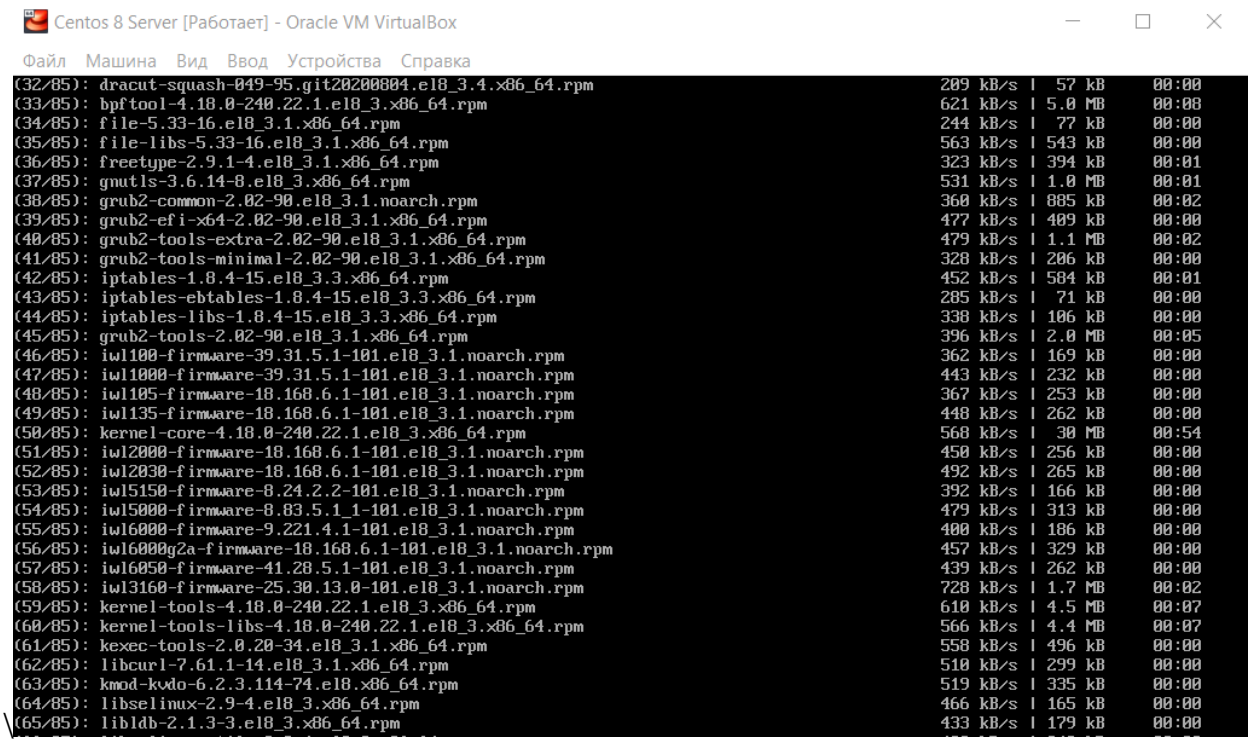


Рисунок 2.4 – Оновлення системи

Додатково завантажимо мережеві утиліти, які можуть знадобитися пізніше, а саме для перегляду налаштувань інтерфейсів, портів, таблиць маршрутизації та інше (рис. 2.5).

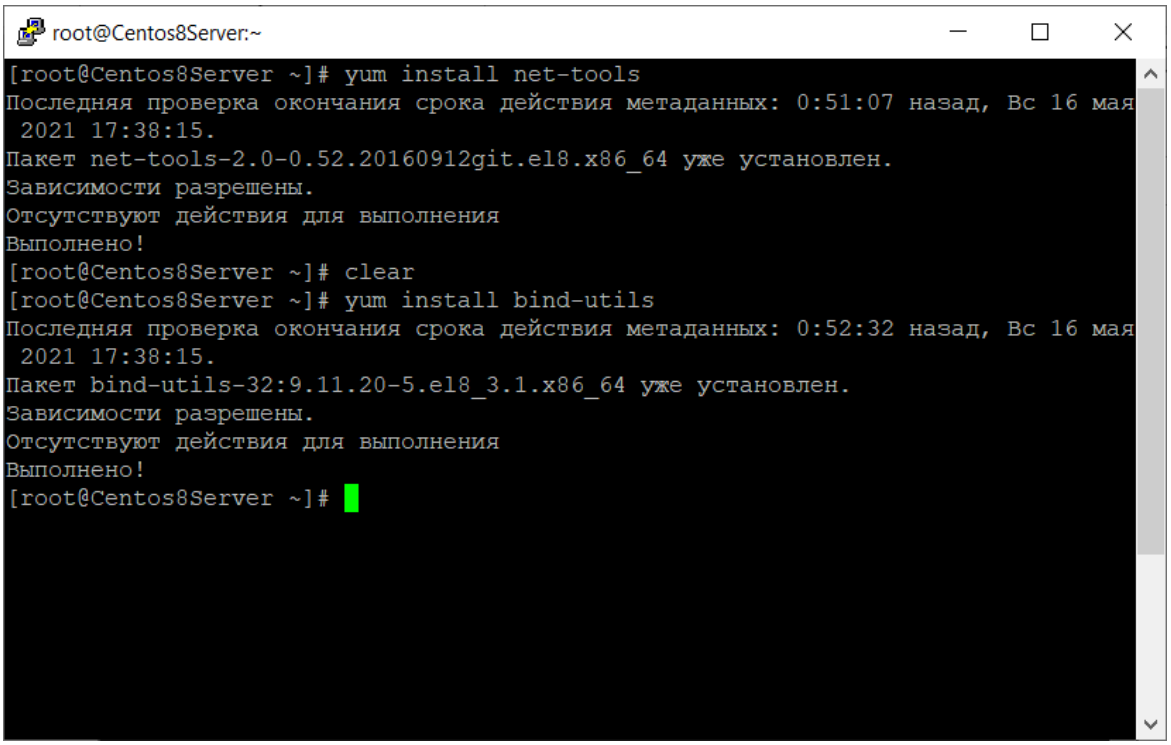


Рисунок 2.5 – Завантаження додаткових утиліт

Тепер завантажимо пакет, за допомогою якого буде налаштовано конфігураційні файли, які відповідають за налаштування мережі та буде підключено додаткові мережеві служби для її керування (рис. 2.6) [7].

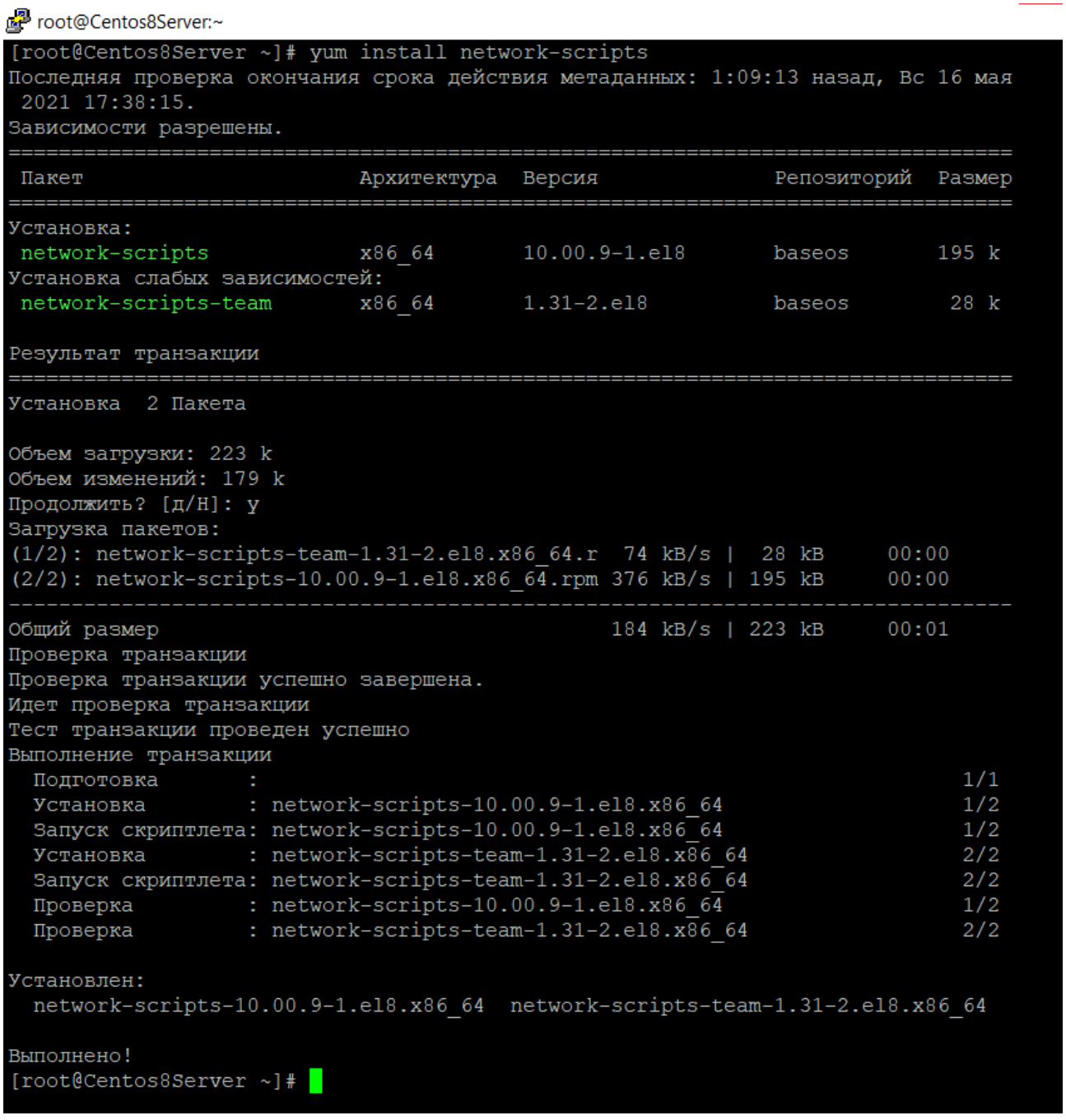


Рисунок 2.6 – Завантаження пакету для налаштування мережі

# 3. СТВОРЕННЯ ВІРТУАЛЬНОГО ЗАХИЩЕНОГО СЕРЕДОВИЩА

Віртуальне середовище буде являти систему, де буде реалізовано безпечну передачу даних за рахунок використання VPN, а саме однієї із реалізацій даної технології – OpenVPN, а також використанні операційної системи Centos 8 та додаткових утиліт. Це дозволить об'єднати локальні мережі та передавати дані поверх Інтернет мережі.

## 3.1 Налаштування брандмауера Centos 8 на шлюзах

Оскільки систему було вже оновлено та завантажено додаткові утиліти для роботи з мережею, почнемо налаштування firewall, який потім буде використовуватися при налаштуванні серверу та клієнту OpenVPN. Для налаштування firewall будемо використовувати утиліту iptables – це більш старий спосіб за допомогою якого можна керувати правилами для трафіку. Також можна використовувати firewalld та nftables – це більш новий інструментарій. Спочатку зупинимо роботу firewalld, який встановлений за замовчуванням, завантажимо iptables (рис. 3.1) та занесемо налаштування в конфігураційний файл /etc/sysconfig/iptables (додаток А) [8].

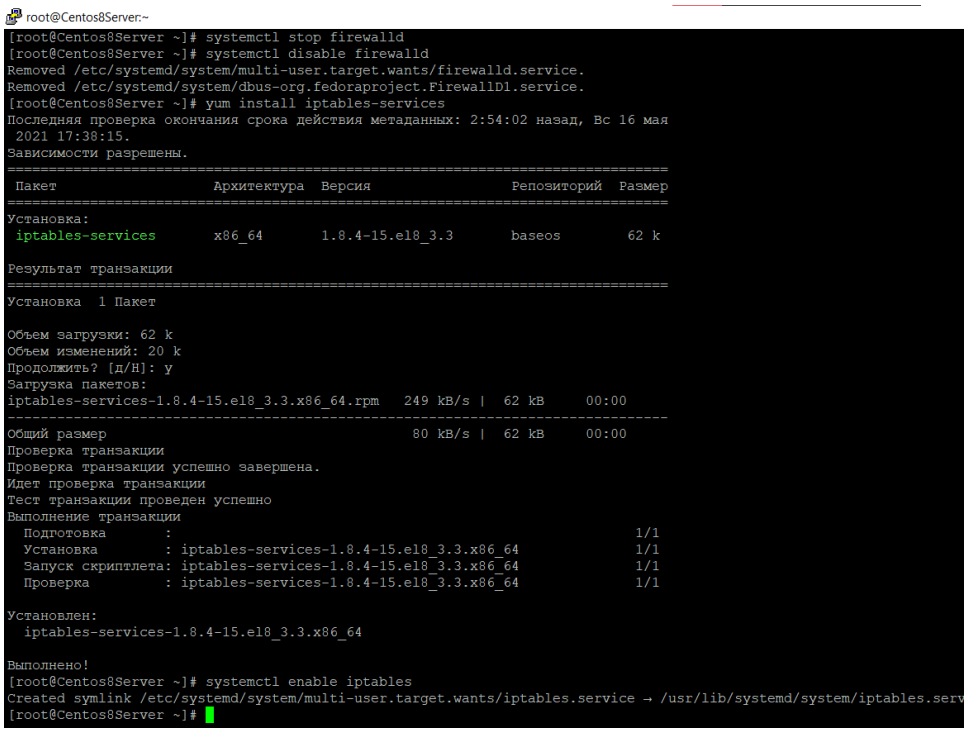


Рисунок 3.1 – Завантаження та підключення сервісу iptables

## 3.2 Налаштування шлюзів у різних сегментах мережі

Також для коректної роботи необхідно відключити SE-Linux на сервері та клієнті. Для того, щоб це зробити необхідно перейти в папку /etc/sysconfig/selinux та змінити значення в рядку SELINUX на disabled та перезапустити систему. SE-Linux – це система контролю, яка реалізується на рівні ядра (рис. 3.2) [5].

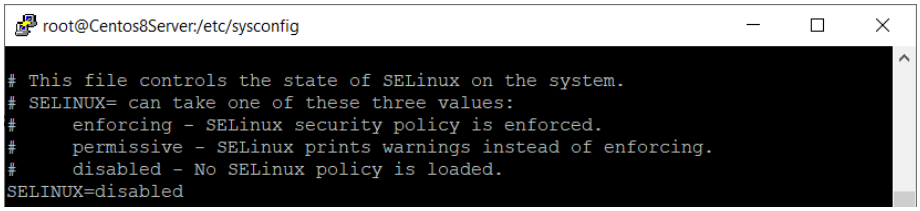


Рисунок 3.2 – Відключення системи контролю

Оскільки комп'ютери на яких знаходяться серверне та клієнтське програмне забезпечення є шлюзами, то необхідно налаштувати їх на проходження трафіку. Для того, щоб це зробити необхідно в конфігураційному файлі /etc/sysctl.conf дописати рядок net.ipv4.ip_forward = 1 (рис. 3.3). Після цього ввести команду sysctl -p для того, щоб налаштування з файлу виконалися. Також для того, щоб наші шлюзи коректно працювали, необхідно додати правило в iptables, тобто в наш firewall. Правило має вигляд - -A POSTROUTING -s 10.0.9.0/24 -o enp0s8 -j MASQUERADE(додаток А). За допомогою даного правила буде відбуватися заміна IP-адреси джерела в заголовці пакету вихідного трафіка на IP-адресу, яка знаходиться на інтерфейсі enp0s8, тобто буде реалізовано технологію NAT[5].

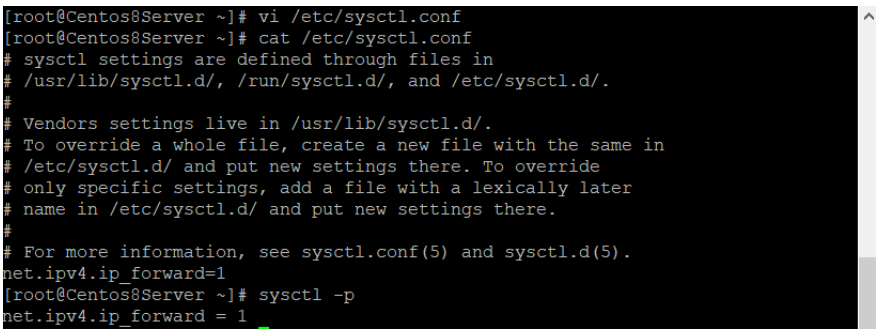


Рисунок 3.3 – Налаштування шлюзів

### 3.3 Налаштування серверу OpenVPN

Тепер безпосередньо перейдемо до завантаження та налаштування серверу OpenVPN. Першим кроком підключимо репозиторій epel-release до нашої операційної системи за допомогою команди yum install epel-release та завантажимо безпосередньо OpenVPN (рис. 3.4, рис. 3.5).

```
================================================================================
 Пакет                Архитектура     Версия             Репозиторий     Размер
================================================================================
Установка:
 epel-release         noarch          8-8.el8            extras           23 k

Результат транзакции
================================================================================
Установка  1 Пакет

Объем загрузки: 23 k
Объем изменений: 32 k
Продолжить? [д/Н]: y
Загрузка пакетов:
epel-release-8-8.el8.noarch.rpm                103 kB/s |  23 kB     00:00
--------------------------------------------------------------------------------
Общий размер                                    34 kB/s |  23 kB     00:00
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно
Выполнение транзакции
  Подготовка       :                                                        1/1
  Установка        : epel-release-8-8.el8.noarch                            1/1
  Запуск скриптлета: epel-release-8-8.el8.noarch                            1/1
  Проверка         : epel-release-8-8.el8.noarch                            1/1

Установлен:
  epel-release-8-8.el8.noarch

Выполнено!
[root@Centos8Server ~]#
```

Рисунок 3.4 – Підключення репозиторію

```
Выполнение транзакции
  Подготовка       :                                                        1/1
  Установка        : pkcs11-helper-1.22-7.el8.x86_64                        1/2
  Запуск скриптлета: openvpn-2.4.11-1.el8.x86_64                            2/2
  Установка        : openvpn-2.4.11-1.el8.x86_64                            2/2
  Запуск скриптлета: openvpn-2.4.11-1.el8.x86_64                            2/2
  Проверка         : openvpn-2.4.11-1.el8.x86_64                            1/2
  Проверка         : pkcs11-helper-1.22-7.el8.x86_64                        2/2

Установлен:
  openvpn-2.4.11-1.el8.x86_64          pkcs11-helper-1.22-7.el8.x86_64

Выполнено!
[root@Centos8Server ~]#
```

Рисунок 3.5 – Підключення репозиторію

Щоб працювати з OpenVPN необхідно спочатку створити сертифікати за допомогою утиліти Easy-RSA, яка є у вільному доступі. Для зберігання ключів створимо директорію mkdir /etc/openvpn/keys. Після цього перейдемо до створеного каталогу, завантажимо там нашу утиліту та розархівуємо завантажений архів (рис. 3.6, рис. 3.7).

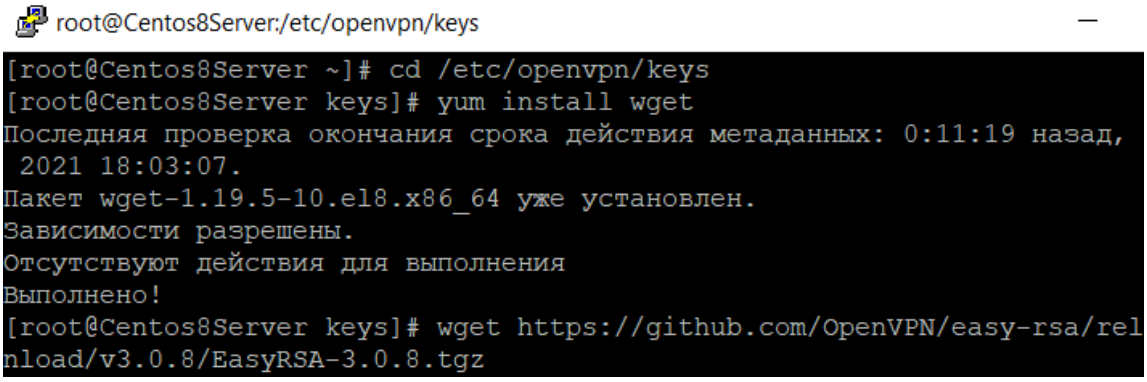


Рисунок 3.6 – Створення сертифікатів

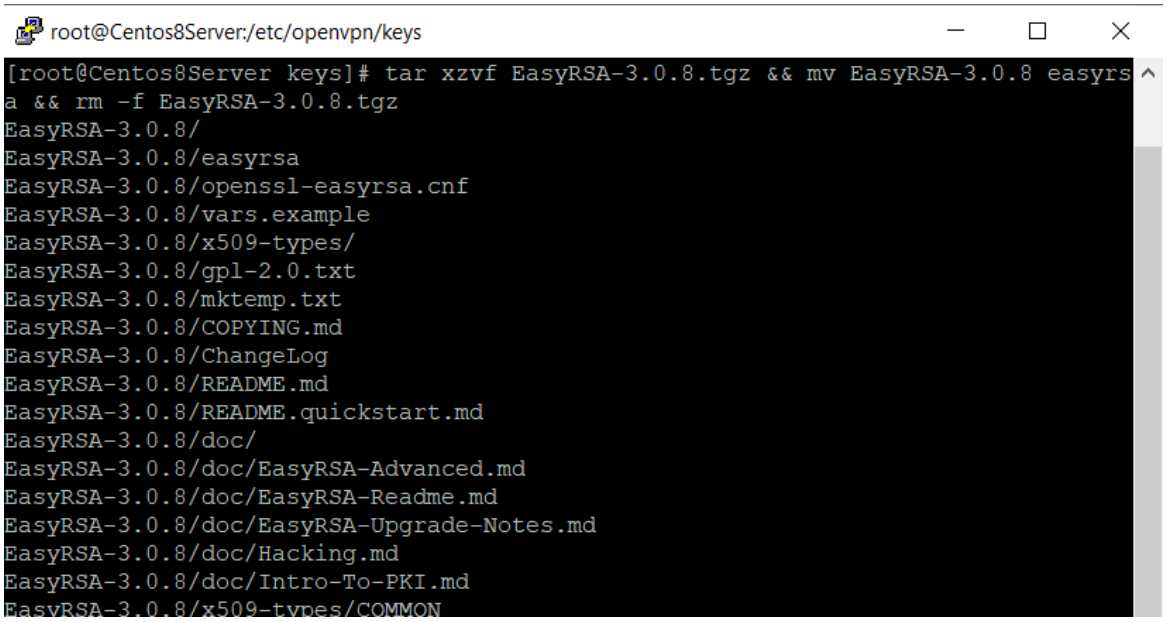


Рисунок 3.7 – Створення сертифікатів

Перейдемо до каталогу /etc/openvpn/keys/easyrsa та створимо структуру публічних ключів за допомогою готового скрипту (рис. 3.8).

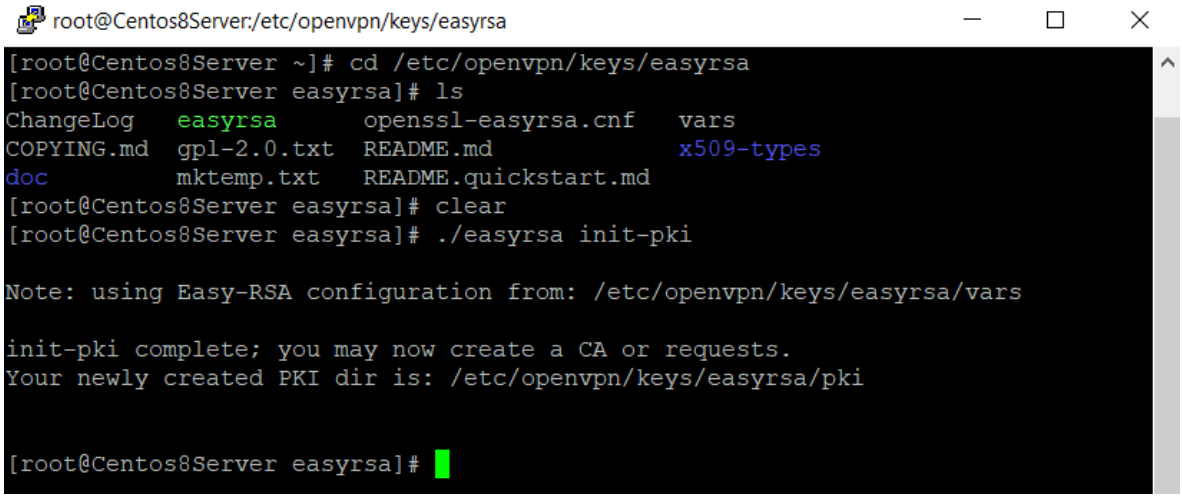


Рисунок 3.8 – Створення публічних ключів

Тепер створимо центр сертифікації. Він потрібен для того, щоб можна було перевірити правдивість ключів шифрування за допомогою сертифікатів з електронним підписом. Обов'язково при створенні центру сертифікації необхідно створити пароль та запам'ятати його, оскільки він знадобиться при створенні нового сертифікату OpenVPN (рис. 3.9).

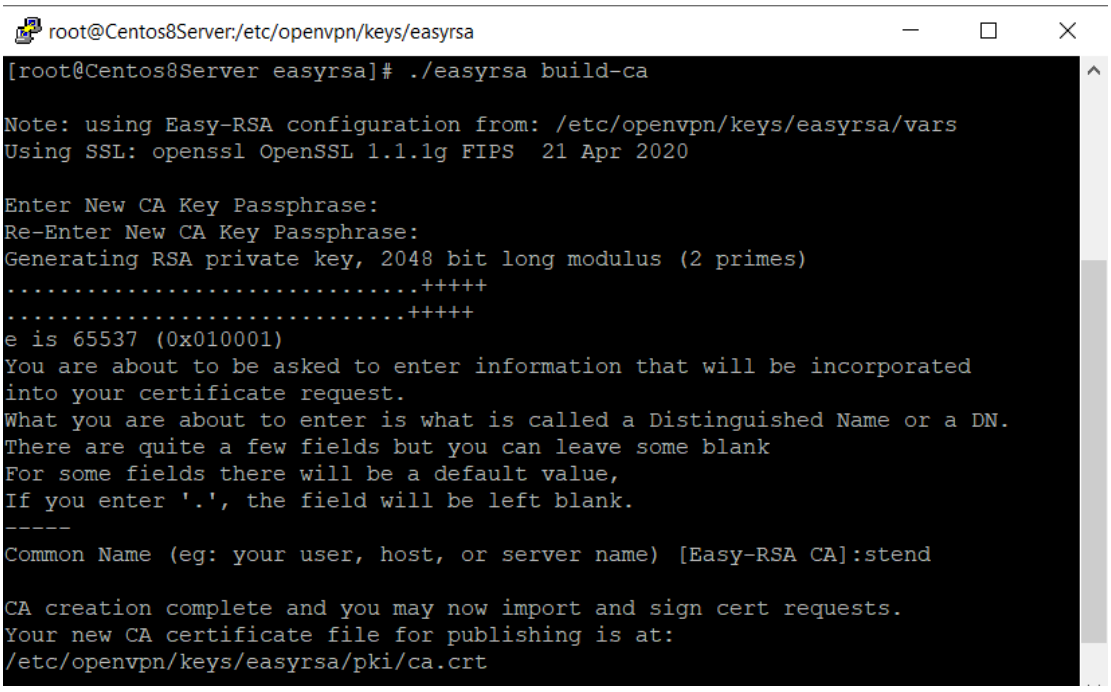


Рисунок 3.9 – Створення центру сертифікації

Після виконаних дій маємо два файли. Перший файл – це приватний або секретний ключ, який завжди зберігається на сервері та нікому не передається. Він зберігається за таким шляхом /etc/openvpn/keys/easyrsa/pki/private/ca.key. Другий ключ – це публічний ключ, який буде передаватися клієнтам разом із призначеними сертифікатами. Він зберігається за таким шляхом /etc/openvpn/keys/easyrsa/pki/ca.crt.

Тепер для зручності користування сервером OpenVPN виконаємо команду ./easyrsa gen-req server nopass. За допомогою цієї команди створимо запит сертифікату для серверу без пароля. Це дозволить запускати сервер та не вводити пароль кожного разу (рис. 3.10).

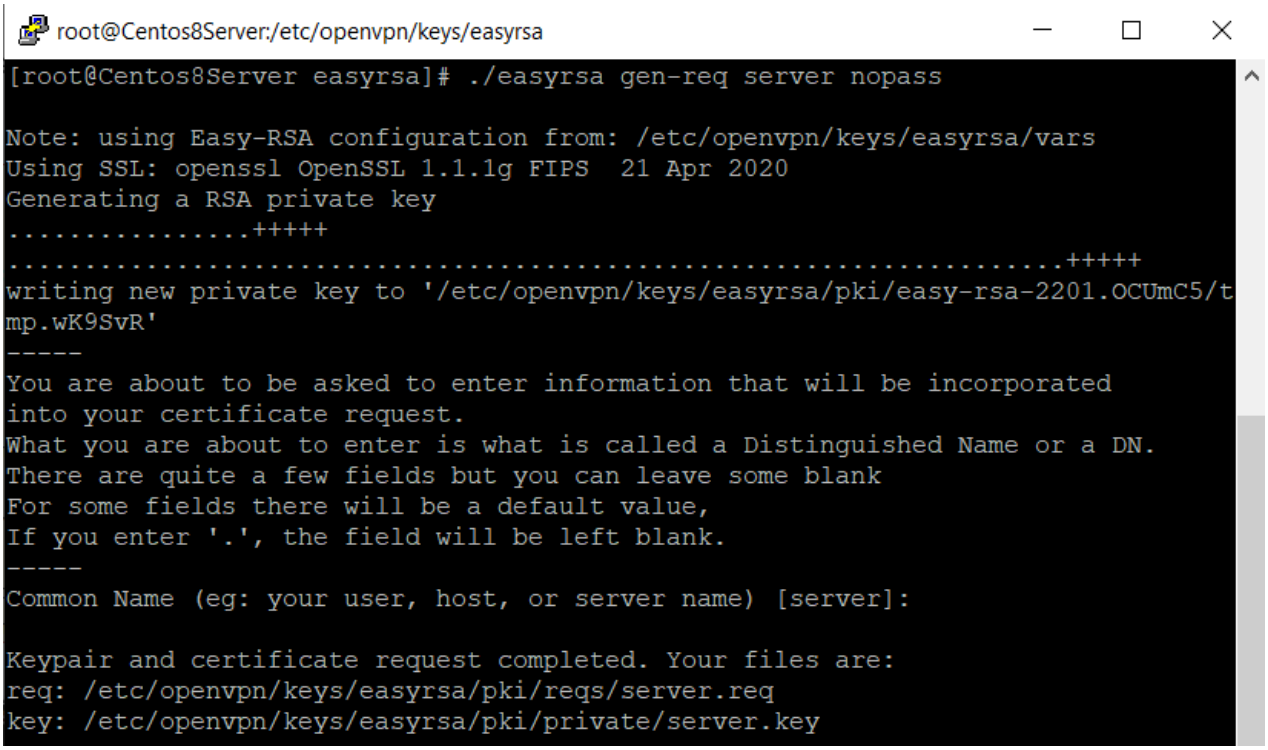


Рисунок 3.10 – Запит сертифікату для серверу без пароля

Тепер підпишемо запит на отримання сертифікату від нашого центру сертифікації за допомогою команди - ./easyrsa sign-req server server (рис. 3.11).

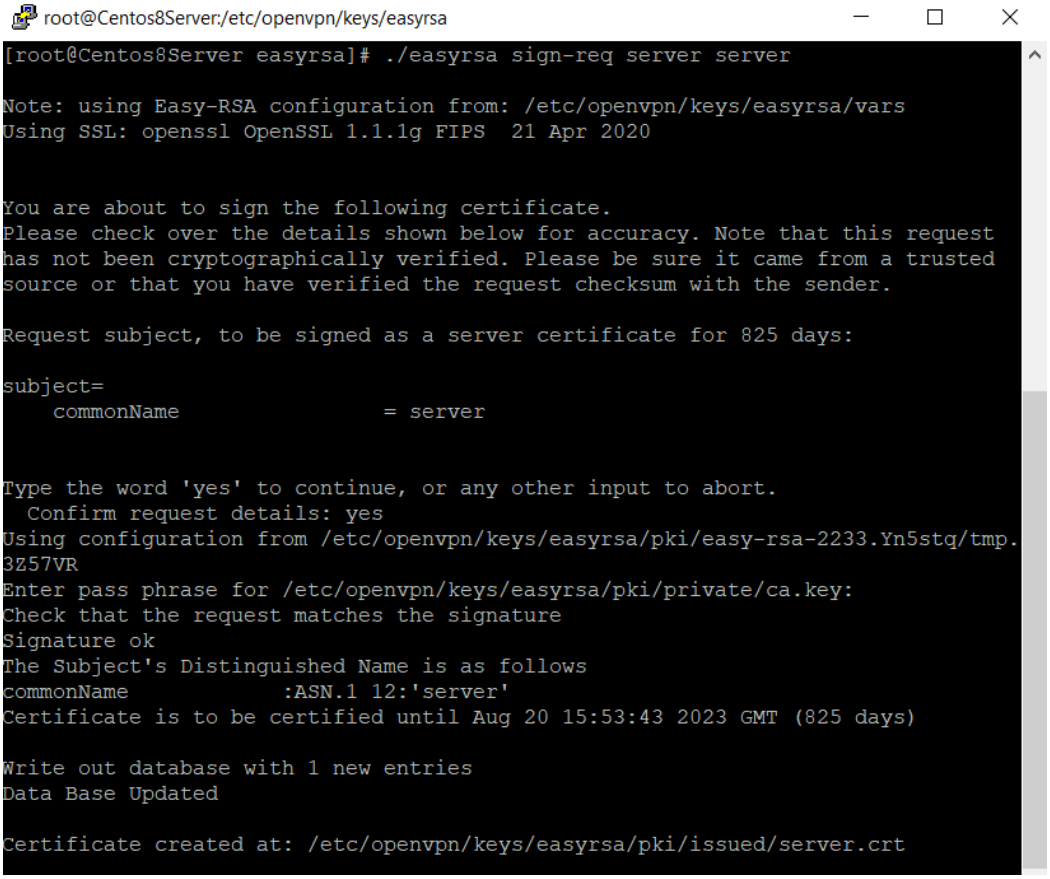


Рисунок 3.11 – Запит на отримання сертифікату

Також для роботи нам знадобиться створити ключ Діффі-Геллмана. Даний ключ буде використовуватися для обміну раніше створених ключів, оскільки даний метод дозволить користувачам обмінятися ключем, який в подальшому буде використовуватися для симетричного шифрування. Даний алгоритм не буде брати участі у шифруванні повідомлень, його основна мета – це розподіл ключів. Після завершення роботи скрипту, буде отримано файл - /etc/openvpn/keys/easy-rsa/pki/dh.pem (рис. 3.12).

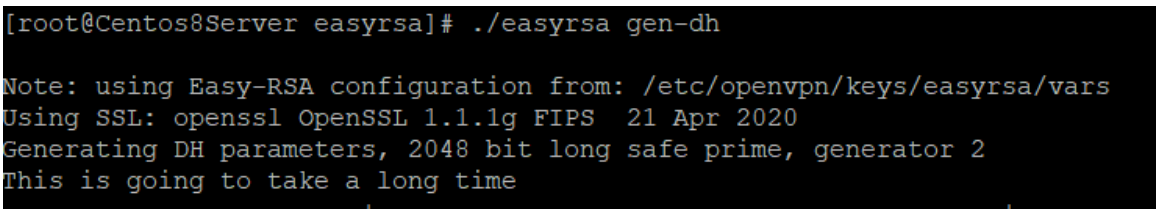


Рисунок 3.12 – Ключ Діффі-Геллмана

Тепер скопіюємо всі необхідні для роботи OpenVPN Server файли в директорію /etc/openvpn/server/ (рис. 3.13).

```
[root@Centos8Server easyrsa]# cp pki/ca.crt /etc/openvpn/server/ca.crt
[root@Centos8Server easyrsa]# cp pki/dh.pem /etc/openvpn/server/dh.pem
[root@Centos8Server easyrsa]# cp pki/issued/server.crt /etc/openvpn/server/serve
r.crt
[root@Centos8Server easyrsa]# cp pki/private/server.key /etc/openvpn/server/serv
er.key
[root@Centos8Server easyrsa]# ls -l /etc/openvpn/server/
итого 20
-rw------- 1 root root 1180 мая 17 19:17 ca.crt
-rw------- 1 root root  424 мая 17 19:17 dh.pem
-rw------- 1 root root 4579 мая 17 19:18 server.crt
-rw------- 1 root root 1708 мая 17 19:19 server.key
[root@Centos8Server easyrsa]#
```

Рисунок 3.13 – Копіювання файлів

Для безпечної роботи необхідно створити також TLS ключ. За допомогою даного ключа буде відбуватися аутентифікація клієнтів, які будуть підключатися до сервера. Даний ключ створюється лише в єдиному екземплярі та повинен знаходитися на кожному комп'ютері, який є учасником VPN-мережі. Для створення TLS ключа необхідно виконати команду openvpn --genkey --secret /etc/openvpn/server/tc.key (рис. 3.14).

root@Centos8Server:/etc/openvpn/keys/easyrsa

```
[root@Centos8Server easyrsa]# openvpn --genkey --secret /etc/openvpn/server/tc.k
ey
[root@Centos8Server easyrsa]# ls -l /etc/openvpn/server/
итого 24
-rw------- 1 root root 1180 мая 17 19:17 ca.crt
-rw------- 1 root root  424 мая 17 19:17 dh.pem
-rw------- 1 root root 4579 мая 17 19:18 server.crt
-rw------- 1 root root 1708 мая 17 19:19 server.key
-rw------- 1 root root  636 мая 17 19:25 tc.key
[root@Centos8Server easyrsa]#
```

Рисунок 3.14 – Створення TLS ключа

Створимо ключ для клієнта OpenVPN за допомогою команд ./easyrsa gen-req client01 nopass та ./easyrsa sign-req client client01 (рис. 3.15, рис. 3.16).

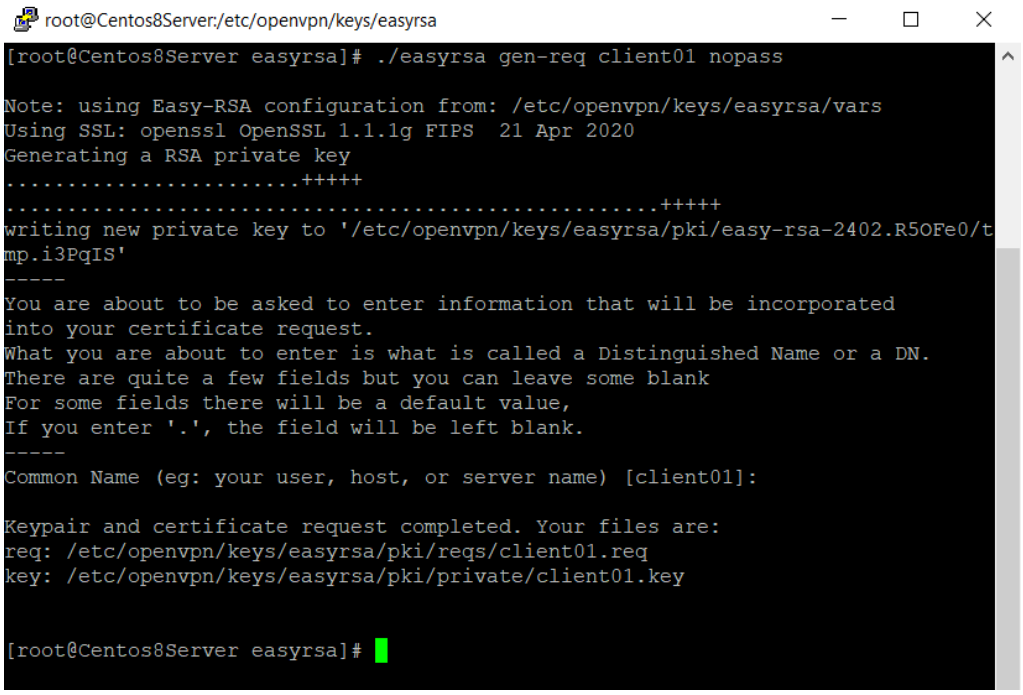


Рисунок 3.15 – Створення ключа для клієнта

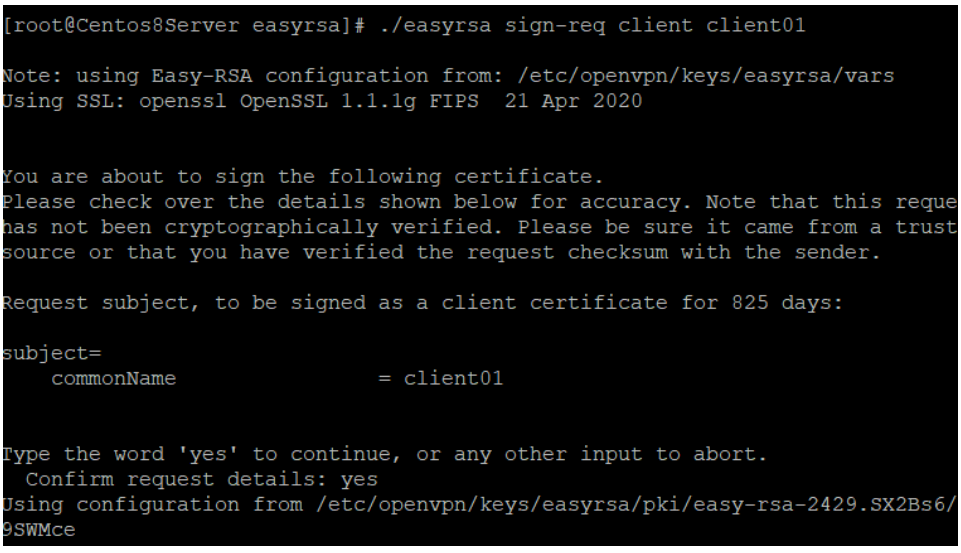


Рисунок 3.16 – Створення ключа для клієнта

Для того, щоб клієнт став частиною мережі VPN, необхідно передати йому набір файлів, а саме client01.crt, client01.key, ca.crt, tc.key на комп’ютер на якому буде знаходитися клієнтська частина.

### 3.4 Налаштування конфігураційного файлу OpenVPN

Першим кроком створимо конфігураційний файл OpenVPN за допомогою команди touch /etc/openvpn/server/server.conf та занесемо в нього такі налаштування (рис. 3.17)[10]:

port 13555 # порт на якому буде працювати VPN

proto udp # протокол, який буде використовуватися на транспортному рівні

dev tun # у нашому випадку ми використовуємо інтерфейс, оскільки ми хочемо об’єднати дві різні локальні мережі для взаємного доступу до даних, а інакше tap.

ca /etc/openvpn/server/ca.crt

cert /etc/openvpn/server/server.crt

key /etc/openvpn/server/server.key

dh /etc/openvpn/server/dh.pem

auth SHA256 # алгоритм шифрування для аутентифікації

cipher AES-256-CBC # алгоритм за допомогою якого будуть шифруватися дані при передачі через vpn.

tls-version-min 1.2 # версія протоколу TLS.

tls-crypt tc.key # ключ для шифрування TLS.

server 10.0.0.0 255.255.255.0 # підмережа для тунелю

route 10.0.10.0 255.255.255.0 # тут вказуємо підмережу до якої будуть надходити запити через VPN.

ifconfig-pool-persist ipp.txt # файл для зберігання відповідностей client-ip.

client-to-client # даний рядок дозволяє користувачам підключатися один до одного.

client-config-dir /etc/openvpn/ccd # директорія для зберігання налаштувань клієнтів.

keepalive 10 120

comp-lzo

explicit-exit-notify

persist-key

persist-tun

status /var/log/openvpn/openvpn-status.log

log /var/log/openvpn/openvpn.log

user nobody

group nobody

verb 3

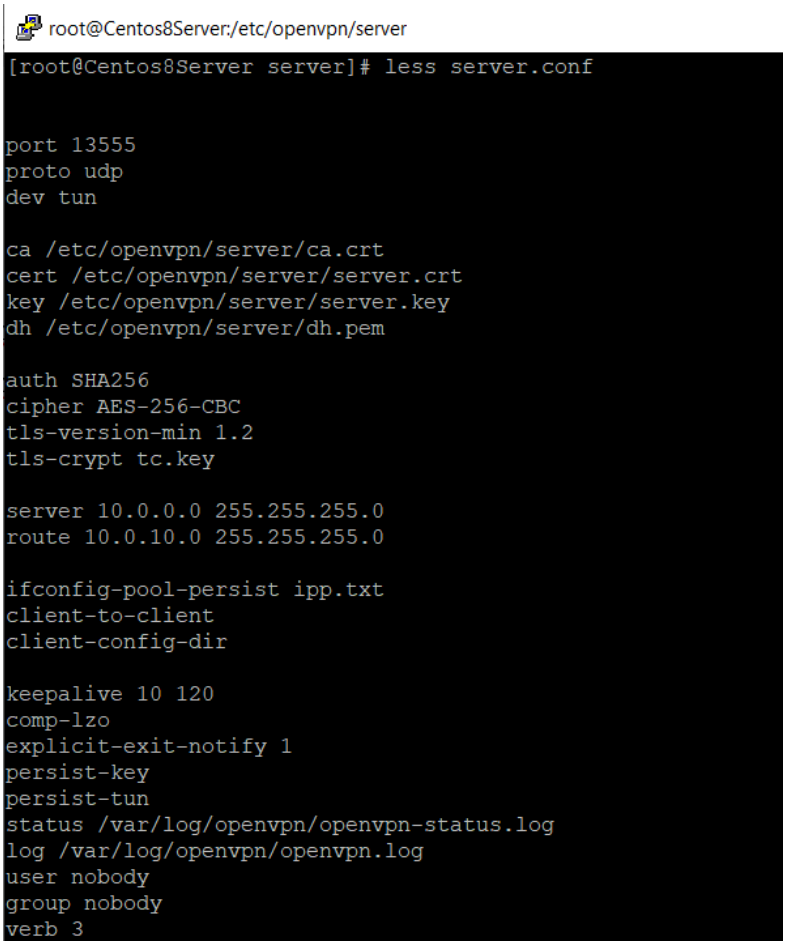


Рисунок 3.17 – Створення конфігураційного файлу OpenVPN

Створимо необхідні директорії для зберігання налаштувань користувачів та для зберігання log файлів (рис. 3.18).

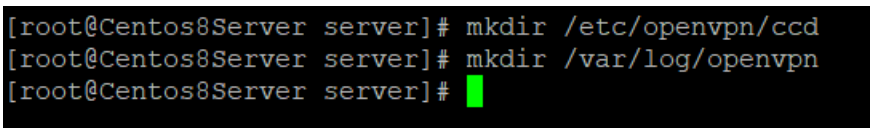


Рисунок 3.18 – Створення додаткових директорій

Тепер в папці де будуть зберігатися конфігураційні файли клієнтів, створимо файл client01 та занесемо в нього параметр iroute 10.0.10.0 255.255.255.0. Даний параметр потрібен для того, щоб співвіднести клієнта та підмережу за яку він відповідає. Зробимо це за допомогою команди touch /etc/openvpn/ccd/client01 (рис. 3.19).

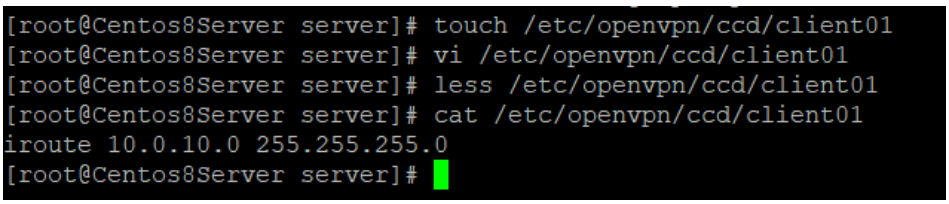


Рисунок 3.19 – Занесення параметрів у файл client01

Після цього можемо запускати наш сервер за допомогою команди systemctl start openvpn-server@server.service та перевіримо статус за допомогою команди systemctl status openvpn-server@server.service (рис. 3.20).

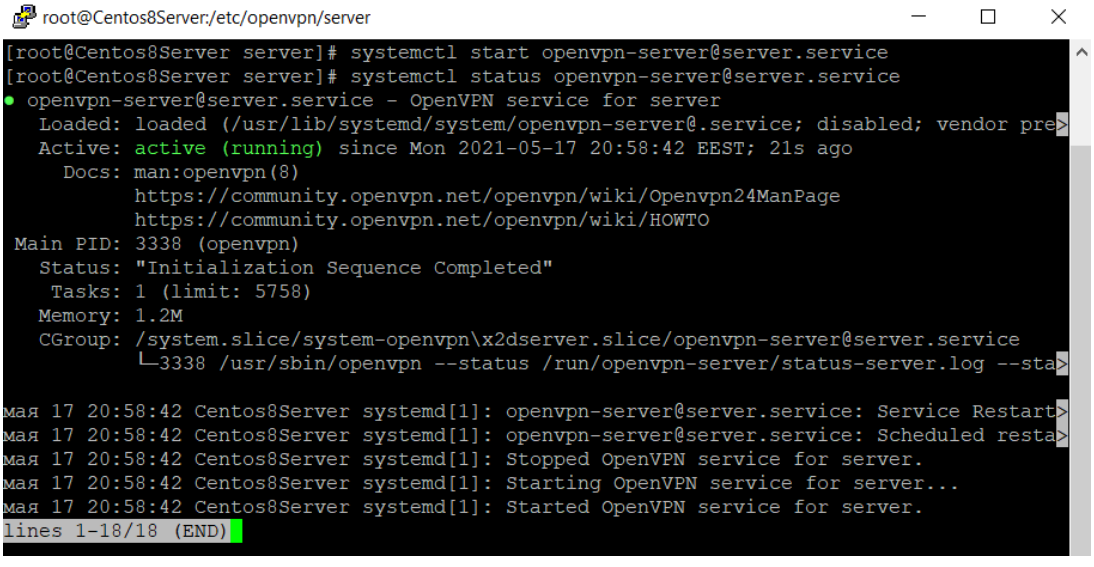


Рисунок 3.20 – Запуск сервера

Перевіримо також чи запустився наш сервер на тому порті, на якому ми зазначили за допомогою команди netstat -tulnp | grep 13555 та додамо наш сервер до автозапуску (рис. 3.21).

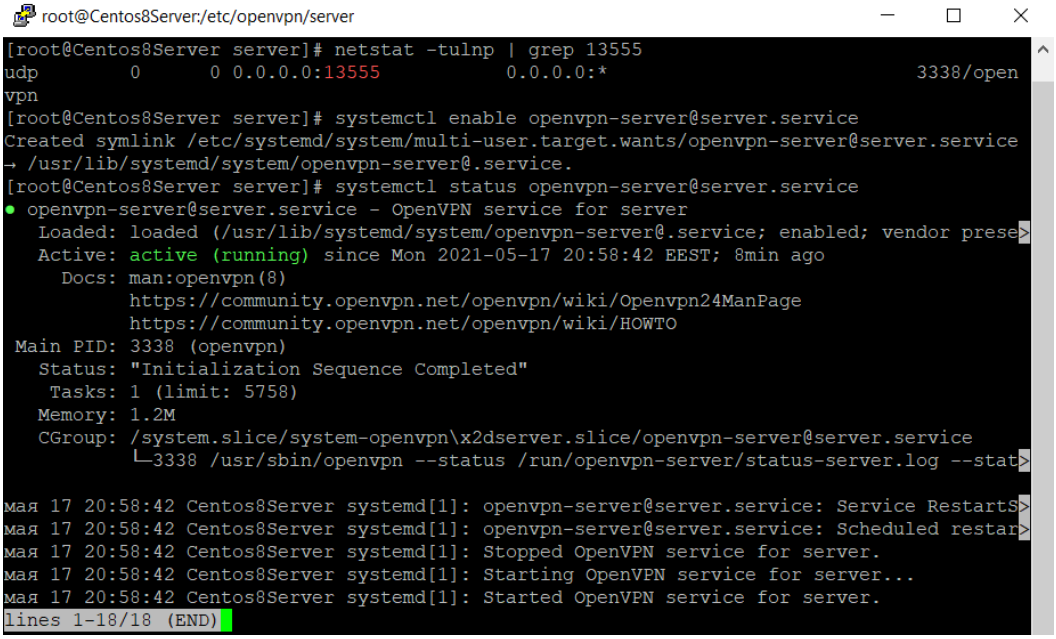


Рисунок 3.21 – Перевірка порту сервера

Тепер перевіримо налаштування мережевих налаштувань, а саме інтерфейсів та таблицю маршрутизації(рис. 3.22).

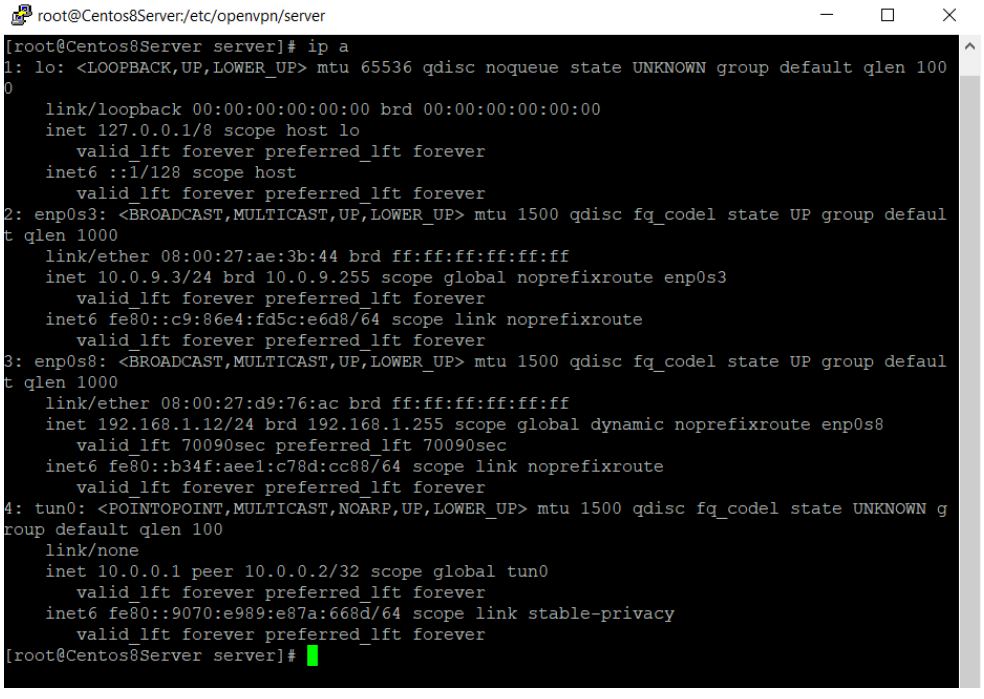


Рисунок 3.22 – Перевірка мережевих налаштувань

В налаштуваннях мережі з’явився новий інтерфейс, а саме tun0, який буде брати участь в утворенні VPN тунелю. Перевіримо таблицю маршрутизації (рис. 3.23).

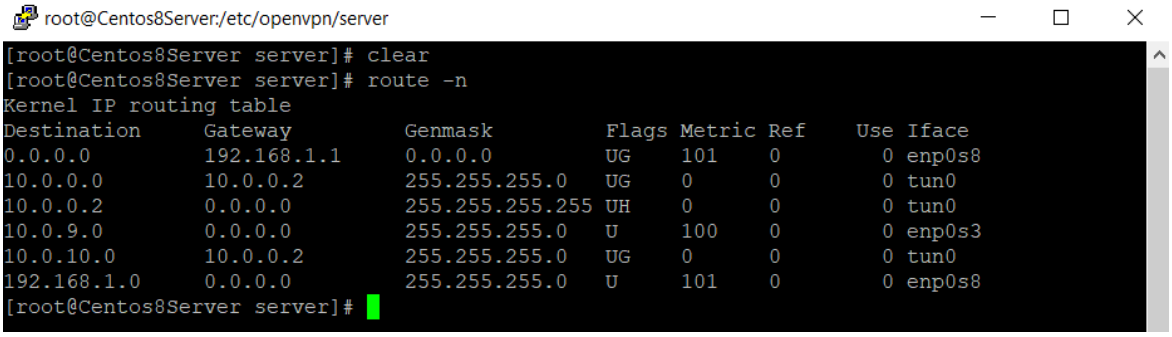


Рисунок 3.23 – Перевірка таблиці маршрутизації

З рисунку бачимо, що трафік мережі 10.0.10.0/24 буде маршрутизуватися в тунель. Важливим пунктом для правильної роботи VPN є налаштування iptables, необхідно додати такі правила, щоб трафік VPN був дозволений в мережі:

iptables -A INPUT -i enp0s8 -p udp --dport 13555 -j ACCEPT

iptables -A INPUT -i tun+ -j ACCEPT

iptables -A OUTPUT -o tun+ -j ACCEPT

iptables -A FORWARD -i tun+ -j ACCEPT

$IPT -t nat -A POSTROUTING -s 10.0.0.0/24 -j MASQUERADE

### 3.5 Налаштування OpenVPN клієнта

Оскільки раніше вже було налаштувано клієнтський комп'ютер як шлюз, правильно налаштовано iptables (додаток А) та відключено SeLinux, то відразу переходимо до завантаження OpenVPN ( рис. 3.24).

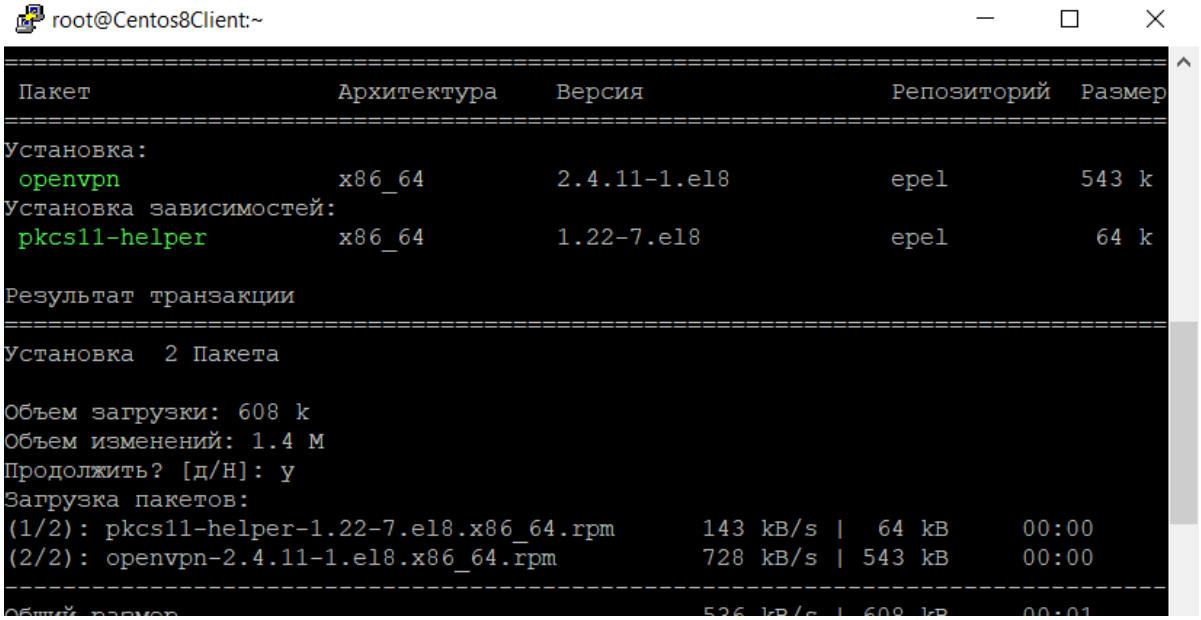


Рисунок 3.24 – Завантаження OpenVPN

Створимо конфігураційний файл в директорії /etc/openvpn/client/ та занесемо в нього такі параметри[10]:

dev tun

proto udp

remote 192.168.1.12 13555

client

resolv-retry infinite

ca /etc/openvpn/client/ca.crt

cert /etc/openvpn/client/client01.crt

key /etc/openvpn/client/client01.key

tls-crypt /etc/openvpn/client/tc.key

route 10.0.9.0 255.255.255.0

remote-cert-tls server

auth SHA256

cipher AES-256-CBC

persist-key

persist-tun

resolv-retry infinite

nobind

comp-lzo

verb 3

status /var/log/openvpn/openvpn-status.log 1

status-version 3

log-append /var/log/openvpn/openvpn-client.log

Також скопіюємо в /etc/openvpn/client/ ключі, які ми згенерували на нашому сервері. Зробимо це за допомогою утиліти WinSCP (рис. 3.25).

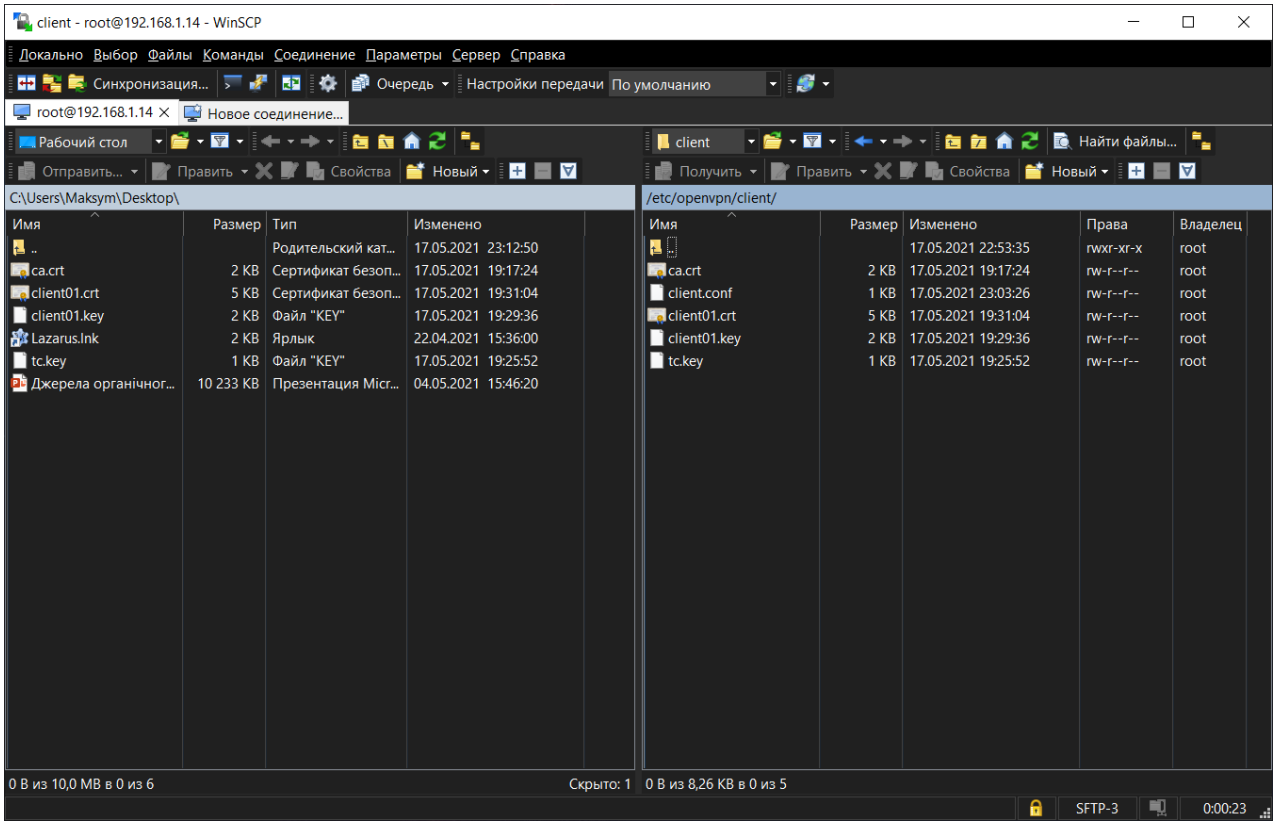


Рисунок 3.25 – Копіювання файлів за допомогою утиліти WinSCP

Створимо директорію для логів /var/log/openvpn, запускаємо клієнтську службу за допомогою команди systemctl start openvpn-client@client.service та перевіримо статус systemctl status openvpn-client@client.service (рис. 3.26).

```
[root@Centos8Client client]# mkdir /var/log/openvpn
[root@Centos8Client client]# systemctl start openvpn-client@client.service
[root@Centos8Client client]# systemctl enable openvpn-client@client.service
Created symlink /etc/systemd/system/multi-user.target.wants/openvpn-client@client.
[root@Centos8Client client]# systemctl status openvpn-client@client.service
● openvpn-client@client.service - OpenVPN tunnel for client
   Loaded: loaded (/usr/lib/systemd/system/openvpn-client@.service; enabled; vendo
   Active: active (running) since Mon 2021-05-17 23:44:44 EEST; 37s ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
 Main PID: 1689 (openvpn)
   Status: "Initialization Sequence Completed"
    Tasks: 1 (limit: 5758)
   Memory: 1.3M
   CGroup: /system.slice/system-openvpn\x2dclient.slice/openvpn-client@client.serv
           └─1689 /usr/sbin/openvpn --suppress-timestamps --nobind --config client

мая 17 23:44:44 Centos8Client systemd[1]: Starting OpenVPN tunnel for client...
мая 17 23:44:44 Centos8Client systemd[1]: Started OpenVPN tunnel for client.
[root@Centos8Client client]#
```

Рисунок 3.26 – Запуск клієнтської служби

Тепер перевіримо налаштування мережевих налаштувань, а саме інтерфейсів та таблицю маршрутизації (рис. 3.27).

```
root@Centos8Client:/etc/openvpn/client
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP grou
p default qlen 1000
    link/ether 08:00:27:a3:a6:5f brd ff:ff:ff:ff:ff:ff
    inet 10.0.10.3/24 brd 10.0.10.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::c41:e877:49f8:9bd8/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP grou
p default qlen 1000
    link/ether 08:00:27:1b:c2:bb brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.14/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0
s8
       valid_lft 81953sec preferred_lft 81953sec
    inet6 fe80::b3a9:d1f8:64e:d4ec/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state U
NKNOWN group default qlen 100
    link/none
    inet 10.0.0.6 peer 10.0.0.5/32 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::81aa:6c81:f748:66df/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
[root@Centos8Client client]#
```

Рисунок 3.27 – Перевірка мережевих налаштувань

В налаштуваннях мережі з’явився новий інтерфейс, а саме tun0, який буде брати участь в утворенні VPN тунелю. Перевіримо таблицю маршрутизації (рис. 3.28).

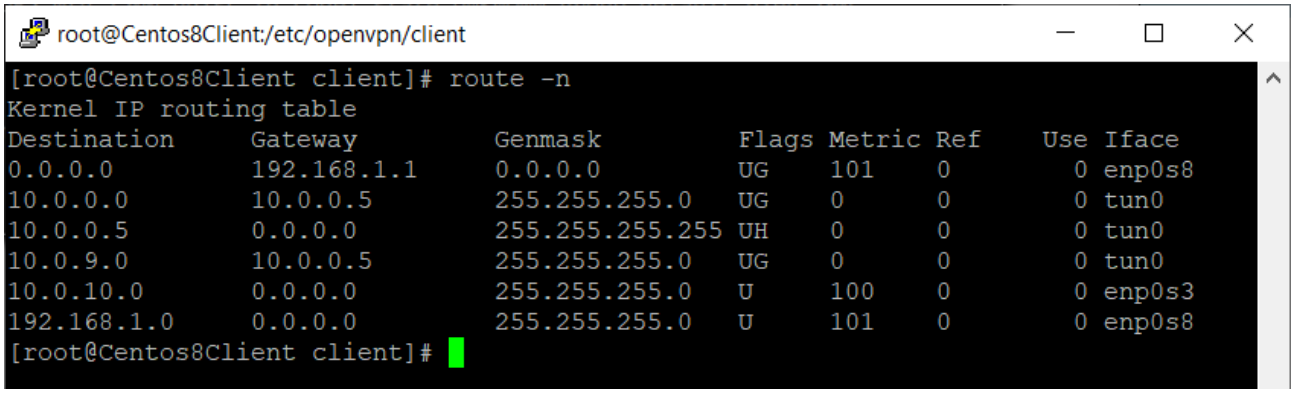


Рисунок 3.28 – Перевірка таблиці маршрутизації

Четверте правило в таблиці чітко демонструє, що трафік до віддаленої мережі 10.0.9.0 буде проходити через VPN-тунель.

### 3.6 Тестування роботи віртуального середовища

Оскільки комп’ютери в приватній мережі уже налаштовані (додаток А), то перевіримо як працює наша VPN мережа. Зайдемо на комп'ютер на якому встановлено клієнтську частину та пропінгуємо сервер – 10.0.9.3 та комп'ютер PC1 – 10.0.9.4 (рис. 3.29).

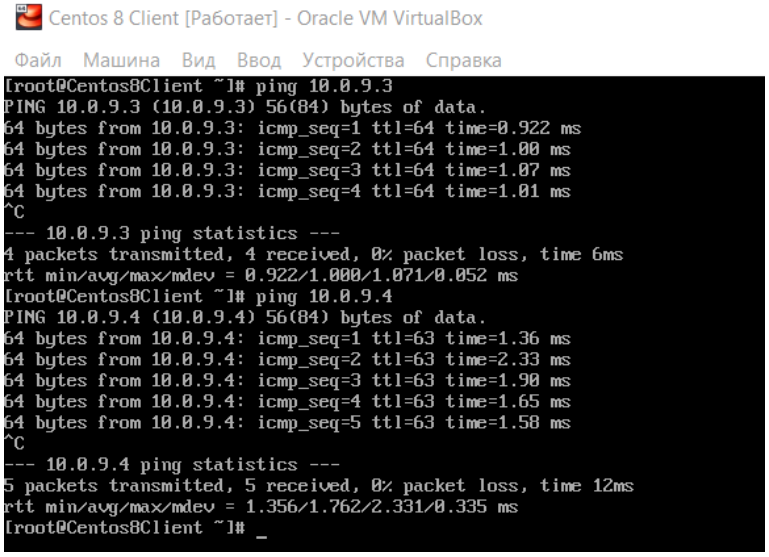


Рисунок 3.29 – Перевірка роботи VPN на Client

Зайдемо на комп'ютер PC2 та пропінгуємо сервер – 10.0.9.3 та комп'ютер PC1 – 10.0.9.4 (рис. 3.30).

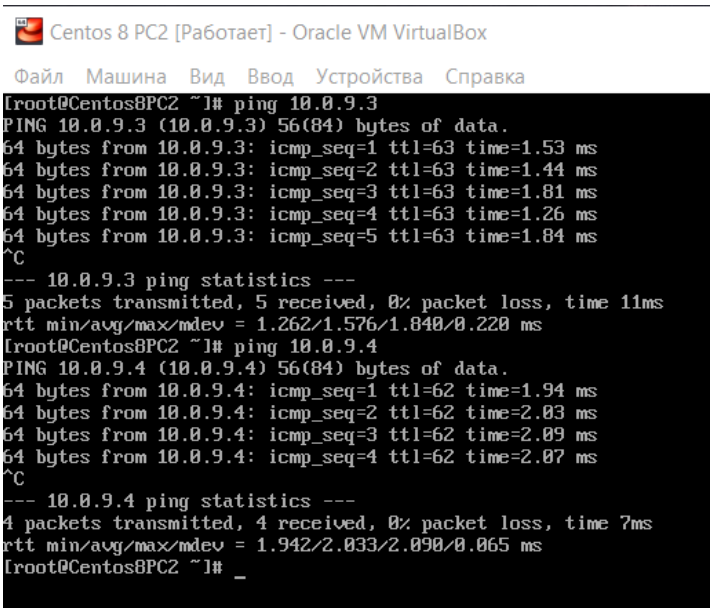


Рисунок 3.30 – Перевірка роботи VPN на PC2

І тепер навпаки зайдемо на комп'ютер PC1 та пропінгуємо комп'ютер, на якому встановлено клієнтську частину – 10.0.10.3, та звичайний комп'ютер PC2 – 10.0.10.4 (рис. 3.31).

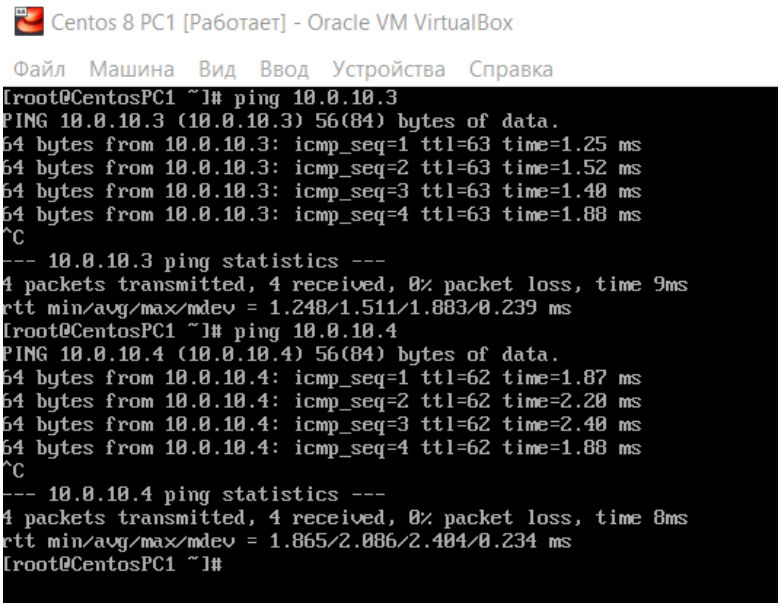


Рисунок 3.31 – Перевірка роботи VPN на PC1

І тепер зайдемо на сервер та пропінгуємо комп’ютери – 10.0.10.3 та 10.0.10.4 (рис. 3.32).

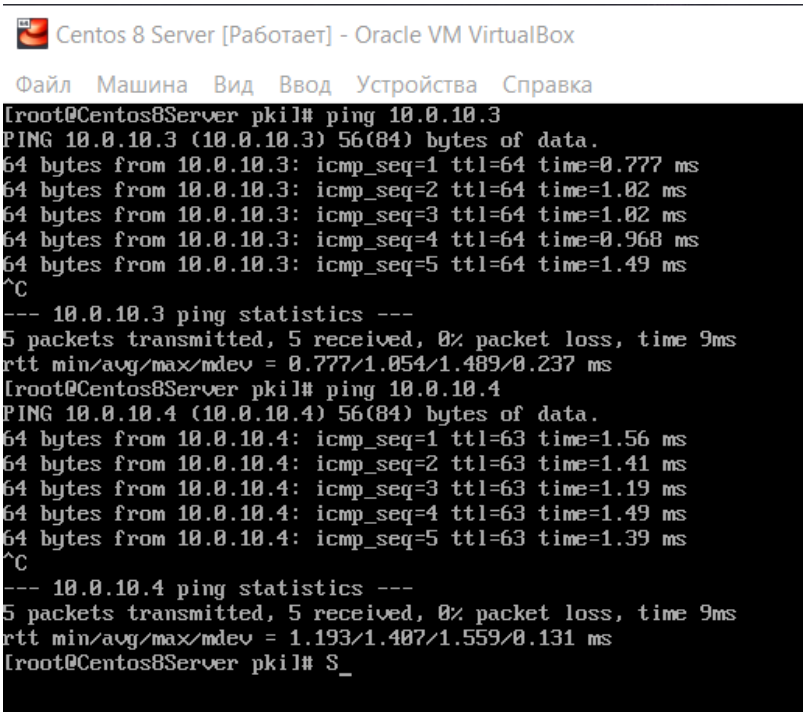


Рисунок 3.32 – Перевірка роботи VPN на Server

# ВИСНОВКИ

У роботі було розглянуто технологію VPN, а саме, переваги використання даної технології в корпоративній мережі, вартість впровадження на підприємстві та зростання популярності на експлуатацію даної технології в майбутньому. Також було розглянуто технологію тунелювання та шифрування пакетів, які задіяні в VPN. Основними компонентами системи VPN, які задіяні в її реалізації, виокремлюють такі: VPN-клієнт, VPN-сервер та шлюз безпеки VPN. Було розглянуто основні протоколи, які задіяні в реалізації технології VPN, такі як PPTP, IPSec, L2TP/IPSec та OpenVPN, а також їх переваги та недоліки при використанні.

Для реалізації поставленої мети було здійснено вибір програмного продукту, який буде задіяний для віртуалізації, операційної системи для побудови віртуального середовища, а також проаналізовані основні програмні додатки, які будуть задіяні при розробці, додаткові репозиторії та конфігураційні файли.

На практичному прикладі було реалізовано віртуальне середовище, за допомогою якого можна детально зрозуміти всі аспекти технології VPN. Було налаштовано усі сегменти мережі, а саме дві локальні мережі та мережа між шлюзами. Під час роботи було завантажено та використано додаткові утиліти, а також було налаштовано брандмауер операційної системи Centos 8, за допомогою якого відбувається керування трафіком. Також було налаштовано серверну та клієнтську частину OpenVPN в різних сегментах мережі, що дало змогу приєднати дві локальні мережі та протестувати правильність роботи створеної системи на кожному пристрої в системі. Дане віртуальне середовище дозволяє безпечно передавати інформацію поверх публічних мереж, з легкістю приєднувати мережі до захищеного середовища, а отже може використовуватися в компаніях, для під’єднання працівників до корпоративної мережі, а також може бути використане для вивчення основних методів побудови захищених з'єднань у мережі.

# СПИСОК ЛІТЕРАТУРИ

1. Олифер В., Олифер Н. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 5-е изд. — СПб.: Питер, 2016. – 996 с.

2. Организация корпоративных сетей на основе VPN построение, управление, безопасность [Електронний ресурс] // Офіційний сайт проекту kp.ru. – 2018 рік – https://www.kp.ru/guide/korporativnaja-set.html.

3. IPSec — протокол защиты сетевого трафика на IP-уровне [Електронний ресурс] // Офіційний сайт проекту ixbt.com. – 2001 рік - https://www.ixbt.com/comm/ipsecure.shtml.

4. Системы защиты Linux [Електронний ресурс] // Офіційний сайт блогу Habr. – 2020 рік – https://habr.com/ru/company/ruvds/blog/523872.

5. Настройка шлюза на CentOS 7 [Електронний ресурс] // Офіційний сайт блогу serveradmin.ru. – 2019 рік – https://serveradmin.ru/nastroyka-shlyuza-dlya-lokalnoy-seti-na-centos-7/.

6. CentOS 7 и 8 настройка сервера после установки [Електронний ресурс] // Офіційний сайт блогу serveradmin.ru. – 2020 рік – https://serveradmin.ru/centos-nastroyka-servera/.

7. Типичная сеть с L2TP [Електронний ресурс] // Офіційний сайт проекту lanmarket.ua. – 2021 рік - https://lanmarket.ua/entsiklopediya/telekommunikatsionnye-tekhnologii/l2tp.html.

8. Use iptables with CentOS 7 [Електронний ресурс] // Офіційний сайт проекту rackspace technology – 2019 рік – https://docs.rackspace.com/support/how-to/use-iptables-with-centos-7/.

9. Шаньгин В.Ф. Информационная безопасность — Москва, 2014. – 702 с.

10. Jan Just Keijser. OpenVPN Cookbook: Second Edition: Birmingham, 2017. – 395 с.

# ДОДАТОК А

**Мережеві налаштування PC1.**

Налаштування файлу /etc/sysconfig/network-scripts/ifcfg-enp0s3

TYPE=Ethernet

PROXY_METHOD=none

BROWSER_ONLY=no

BOOTPROTO=NONE

DEFROUTE=yes

IPADDR=10.0.9.4

PREFIX=24

GATEWAY=192.168.1.12

DNS1=8.8.8.8

IPV4_FAILURE_FATAL=no

NAME=enp0s3

DEVICE=enp0s3

ONBOOT=yes

**Мережеві налаштування PC2.**

Налаштування файлу /etc/sysconfig/network-scripts/ifcfg-enp0s3

TYPE=Ethernet

PROXY_METHOD=none

BROWSER_ONLY=no

BOOTPROTO=NONE

DEFROUTE=yes

IPADDR=10.0.10.4

PREFIX=24

GATEWAY=192.168.1.14

DNS1=8.8.8.8

IPV4_FAILURE_FATAL=no

NAME=enp0s3

DEVICE=enp0s3

ONBOOT=yes

**Налаштування серверу.**

Налаштування файлу /etc/sysconfig/network-scripts/ifcfg-enp0s3

TYPE=Ethernet

PROXY_METHOD=none

BROWSER_ONLY=no

BOOTPROTO=none

DEFROUTE=yes

IPADDR=10.0.9.3

PREFIX=24

IPV4_FAILURE_FATAL=no

NAME=enp0s3

DEVICE=enp0s3

ONBOOT=yes

Налаштування файлу /etc/sysconfig/network-scripts/ifcfg-enp0s8

TYPE=Ethernet

PROXY_METHOD=none

BROWSER_ONLY=no

BOOTPROTO=dhcp

DEFROUTE=yes

IPV4_FAILURE_FATAL=no

NAME=enp0s8

DEVICE=enp0s8

ONBOOT=yes

Налаштування брандмауера сервера.

*raw

```
:PREROUTING ACCEPT
:OUTPUT ACCEPT
COMMIT
*filter
:INPUT DROP
:FORWARD DROP
:OUTPUT DROP
-A INPUT -i lo -j ACCEPT
-A INPUT -i enp0s3 -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 11 -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -m state --state INVALID -j DROP
-A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG NONE -j DROP
-A INPUT -p tcp -m tcp ! --tcp-flags FIN,SYN,RST,ACK SYN -m state --state NEW -j DROP
-A INPUT -i tun+ -j ACCEPT
-A INPUT -i enp0s8 -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -i enp0s8 -p udp -m udp --dport 13555 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -m state --state INVALID -j DROP
-A FORWARD -i enp0s3 -o enp0s8 -j ACCEPT
-A FORWARD -i enp0s8 -o enp0s3 -j REJECT --reject-with icmp-port-unreachable
-A FORWARD -i tun+ -j ACCEPT
-A FORWARD -i enp0s3 -o tun+ -j ACCEPT
```

-A OUTPUT -o lo -j ACCEPT

-A OUTPUT -o enp0s3 -j ACCEPT

-A OUTPUT -o enp0s8 -j ACCEPT

-A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

-A OUTPUT -p tcp -m tcp ! --tcp-flags FIN,SYN,RST,ACK SYN -m state --state NEW -j DROP

-A OUTPUT -o tun+ -j ACCEPT

COMMIT

*nat

:PREROUTING ACCEPT

:INPUT ACCEPT

:POSTROUTING ACCEPT

:OUTPUT ACCEP

-A POSTROUTING -s 10.0.0.0/24 -j MASQUERADE

-A POSTROUTING -s 10.0.9.0/24 -o enp0s8 -j MASQUERADE

COMMIT

*mangle

:PREROUTING ACCEPT

:INPUT ACCEPT

:FORWARD ACCEPT

:OUTPUT ACCEPT

:POSTROUTING ACCEPT

COMMIT

**Налаштування ПК з клієнтською частиною.**

Налаштування файлу /etc/sysconfig/network-scripts/ifcfg-enp0s3

TYPE=Ethernet

PROXY_METHOD=none

BROWSER_ONLY=no

BOOTPROTO=none

DEFROUTE=yes

IPADDR=10.0.10.3

PREFIX=24

IPV4_FAILURE_FATAL=no

NAME=enp0s3

DEVICE=enp0s3

ONBOOT=yes

Налаштування файлу /etc/sysconfig/network-scripts/ifcfg-enp0s8

TYPE=Ethernet

PROXY_METHOD=none

BROWSER_ONLY=no

BOOTPROTO=dhcp

DEFROUTE=yes

IPV4_FAILURE_FATAL=no

NAME=enp0s8

DEVICE=enp0s8

ONBOOT=yes

Налаштування брандмауера на ПК з клієнтською частиною.

*filter

:INPUT DROP

:FORWARD DROP

:OUTPUT DROP

-A INPUT -i lo -j ACCEPT

-A INPUT -i enp0s3 -j ACCEPT

-A INPUT -p icmp -m icmp --icmp-type 0 -j ACCEPT

-A INPUT -p icmp -m icmp --icmp-type 3 -j ACCEPT

-A INPUT -p icmp -m icmp --icmp-type 11 -j ACCEPT

-A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT

-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

-A INPUT -m state --state INVALID -j DROP

-A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG NONE -j DROP

-A INPUT -p tcp -m tcp ! --tcp-flags FIN,SYN,RST,ACK SYN -m state --state NEW -j DROP

-A INPUT -i tun+ -j ACCEPT

-A INPUT -i enp0s8 -p tcp -m tcp --dport 22 -j ACCEPT

-A INPUT -i enp0s8 -p udp -m udp --dport 13555 -j ACCEPT

-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT

-A FORWARD -m state --state INVALID -j DROP

-A FORWARD -i enp0s3 -o enp0s8 -j ACCEPT

-A FORWARD -i enp0s8 -o enp0s3 -j REJECT --reject-with icmp-port-unreachable

-A FORWARD -i tun+ -j ACCEPT

-A FORWARD -i enp0s3 -o tun+ -j ACCEPT

-A OUTPUT -o lo -j ACCEPT

-A OUTPUT -o enp0s3 -j ACCEPT

-A OUTPUT -o enp0s8 -j ACCEPT

-A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

-A OUTPUT -p tcp -m tcp ! --tcp-flags FIN,SYN,RST,ACK SYN -m state --state NEW -j DROP

-A OUTPUT -o tun+ -j ACCEPT

COMMIT

*nat

:PREROUTING ACCEPT

:INPUT ACCEPT

:POSTROUTING ACCEPT

:OUTPUT ACCEPT

```
-A POSTROUTING -s 10.0.0.0/24 -j MASQUERADE
-A POSTROUTING -s 10.0.10.0/24 -o enp0s8 -j MASQUERADE
COMMIT
*mangle
:PREROUTING ACCEPT
:INPUT ACCEPT
:FORWARD ACCEPT
:OUTPUT ACCEPT
:POSTROUTING ACCEPT
COMMIT
```